



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

## An Interior Point-Proximal Method of Multipliers for Convex Quadratic Programming

**Citation for published version:**

Poukakiotis, S & Gondzio, J 2020, 'An Interior Point-Proximal Method of Multipliers for Convex Quadratic Programming', *Computational optimization and applications*. <https://doi.org/10.1007/s10589-020-00240-9>

**Digital Object Identifier (DOI):**

[10.1007/s10589-020-00240-9](https://doi.org/10.1007/s10589-020-00240-9)

**Link:**

[Link to publication record in Edinburgh Research Explorer](#)

**Document Version:**

Peer reviewed version

**Published In:**

Computational optimization and applications

**General rights**

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



# An Interior Point-Proximal Method of Multipliers for Convex Quadratic Programming

Spyridon Pougkakiotis · Jacek Gondzio

Received: date / Accepted: date

**Abstract** In this paper we combine an infeasible Interior Point Method (IPM) with the Proximal Method of Multipliers (PMM). The resulting algorithm (IP-PMM) is interpreted as a primal-dual regularized IPM, suitable for solving linearly constrained convex quadratic programming problems. We apply few iterations of the interior point method to each sub-problem of the proximal method of multipliers. Once a satisfactory solution of the PMM sub-problem is found, we update the PMM parameters, form a new IPM neighbourhood and repeat this process. Given this framework, we prove polynomial complexity of the algorithm, under standard assumptions. To our knowledge, this is the first polynomial complexity result for a primal-dual regularized IPM. The algorithm is guided by the use of a single penalty parameter; that of the logarithmic barrier. In other words, we show that IP-PMM inherits the polynomial complexity of IPMs, as well as the strict convexity of the PMM sub-problems. The updates of the penalty parameter are controlled by IPM, and hence are well-tuned, and do not depend on the problem solved. Furthermore, we study the behavior of the method when it is applied to an infeasible problem, and identify a necessary condition for infeasibility. The latter is used to construct an infeasibility detection mechanism. Subsequently, we provide a robust implementation of the presented algorithm and test it over a set of small to large scale linear and convex quadratic programming problems. The numerical results demonstrate the benefits of using regularization in IPMs as well as the reliability of the method.

## 1 Introduction

### 1.1 Primal-Dual Pair of Convex Quadratic Programming Problems

In this paper, we consider the following primal-dual pair of linearly constrained convex quadratic programming problems, in the standard form:

$$\min_x \left( c^T x + \frac{1}{2} x^T Q x \right), \quad \text{s.t. } Ax = b, \quad x \geq 0, \quad (\text{P})$$

$$\max_{x,y,z} \left( b^T y - \frac{1}{2} x^T Q x \right), \quad \text{s.t. } -Qx + A^T y + z = c, \quad z \geq 0, \quad (\text{D})$$

with  $c, x, z \in \mathbb{R}^n$ ,  $b, y \in \mathbb{R}^m$ ,  $A \in \mathbb{R}^{m \times n}$ , and symmetric positive semi-definite  $Q \in \mathbb{R}^{n \times n}$  (i.e.  $Q \succeq 0$ ). Without loss of generality we assume that  $m \leq n$ . Duality, between the problems (P)-(D), arises by introducing the Lagrangian function of the primal, using  $y \in \mathbb{R}^m$  and  $z \in \mathbb{R}^n$ ,  $z \geq 0$ , as the Lagrange multipliers for the equality and inequality constraints, respectively. Hence, we obtain:

$$\mathcal{L}(x, y, z) := c^T x + \frac{1}{2} x^T Q x - y^T (Ax - b) - z^T x. \quad (1)$$

---

S. Pougkakiotis  
University of Edinburgh  
E-mail: s.pougkakiotis@ed.ac.uk

J. Gondzio  
University of Edinburgh  
E-mail: j.gondzio@ed.ac.uk

Using the Lagrangian function, one can formulate the first-order optimality conditions (known as Karush–Kuhn–Tucker (KKT) conditions) for this primal-dual pair. In particular, we define the vector  $w = (x^T, y^T, z^T)^T$ , and compute the gradient of  $\mathcal{L}(w)$ . Using  $\nabla \mathcal{L}(w)$ , as well as the complementarity conditions, we may define a function  $F(w) : \mathbb{R}^{2n+m} \mapsto \mathbb{R}^{2n+m}$ , using which, we write the KKT conditions as follows:

$$F(w) := \begin{bmatrix} c + Qx - A^T y - z \\ Ax - b \\ XZe_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad (x, z) \geq (0, 0), \quad (2)$$

where  $e_n$  denotes the vector of ones of size  $n$ , while  $X, Z \in \mathbb{R}^{n \times n}$  denote the diagonal matrices satisfying  $X^{ii} = x^i$  and  $Z^{ii} = z^i$ , for all  $i \in \{1, \dots, n\}$ . For the rest of this manuscript, superscripts will denote the components of a vector/matrix. For simplicity of exposition, when referring to convex quadratic programming problems, we implicitly assume that the problems are linearly constrained. If both (P) and (D) are feasible problems, it can easily be verified that there exists an optimal primal-dual triple  $(x, y, z)$ , satisfying the KKT optimality conditions of this primal-dual pair (see for example [7, Prop. 2.3.4]).

## 1.2 A Primal-Dual Interior Point Method

Primal-dual Interior Point Methods (IPMs) are popular for simultaneously solving (P) and (D), iteratively. As indicated in the name, primal-dual IPMs act on both primal and dual variables. There are numerous variants of IPMs and the reader is referred to [17] for an extended literature review. In this paper, an infeasible primal-dual IPM is presented. Such methods are called infeasible because they allow intermediate iterates of the method to be infeasible for (P)-(D), in contrast to feasible IPMs, which require intermediate iterates to be strictly feasible.

Interior point methods handle the non-negativity constraints of the problems with logarithmic barriers in the objective. That is, at each iteration, we choose a *barrier parameter*  $\mu$  and form the logarithmic *barrier primal-dual pair*:

$$\min_x \left( c^T x + \frac{1}{2} x^T Q x - \mu \sum_{j=1}^n \ln x^j \right) \quad \text{s.t.} \quad Ax = b, \quad (3)$$

$$\max_{x, y, z} \left( b^T y - \frac{1}{2} x^T Q x + \mu \sum_{j=1}^n \ln z^j \right) \quad \text{s.t.} \quad -Qx + A^T y + z = c, \quad (4)$$

in which non-negativity constraints  $x > 0$  and  $z > 0$  are implicit. We form the KKT optimality conditions of the pair (3)-(4), by introducing the Lagrangian of the primal barrier problem:

$$\mathcal{L}_\mu^{\text{IPM}}(x, y) := c^T x + \frac{1}{2} x^T Q x - y^T (Ax - b) - \mu \sum_{i=1}^n \ln x^i.$$

Equating the gradient of the previous function to zero, gives the following conditions:

$$\begin{aligned} \nabla_x \mathcal{L}_\mu^{\text{IPM}}(x, y) &= c + Qx - A^T y - \mu X^{-1} e_n = 0, \\ \nabla_y \mathcal{L}_\mu^{\text{IPM}}(x, y) &= Ax - b = 0. \end{aligned}$$

Using the variable  $z = \mu X^{-1} e_n$ , the final conditions read as follows:

$$\begin{aligned} c + Qx - A^T y - z &= 0 \\ Ax - b &= 0 \\ XZe_n - \mu e_n &= 0. \end{aligned}$$

At each IPM iteration, we want to approximately solve the following mildly non-linear system:

$$F_{\sigma, \mu}^{\text{IPM}}(w) := \begin{bmatrix} c + Qx - A^T y - z \\ b - Ax \\ XZe_n - \sigma \mu e_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad (5)$$

where  $F_{\sigma,\mu}^{\text{IPM}}(w) = 0$  is a slightly perturbed form of the previously presented optimality conditions. In particular,  $\sigma \in (0, 1)$  is a *centering parameter* which determines how fast  $\mu$  will be forced to decrease at the next IPM iteration. For  $\sigma = 1$ , we recover the barrier optimality conditions, while for  $\sigma = 0$  we recover the initial problems' optimality conditions (2). The efficiency of the method depends heavily on the choice of  $\sigma$ . In fact, various improvements of the traditional IPM schemes have been proposed in the literature, which solve the previous system for multiple carefully chosen values of the centering parameter  $\sigma$  and of the right hand side, at each IPM iteration. These are the so called *predictor-corrector* schemes, proposed for the first time in [23]. Various extensions of such methods have been proposed and analyzed in the literature (e.g. see [16, 35] and references therein). However, for simplicity of exposition, we will follow the traditional approach;  $\sigma$  is chosen heuristically at each iteration, and the previous system is solved only once.

In order to approximately solve the system  $F_{\sigma,\mu}^{\text{IPM}}(w) = 0$  for each value of  $\mu$ , the most common approach is to apply Newton's method. Newton's method is favored for systems of this form, due to the *self-concordance* of the function  $\ln(\cdot)$ . For more details on this subject, the interested reader is referred to [25, Chapter 2]. At the beginning of the  $k$ -th iteration of the IPM, for  $k \geq 0$ , we have available an iterate  $w_k = (x_k^T, y_k^T, z_k^T)^T$ , and a barrier parameter  $\mu_k$ , defined as  $\mu_k = \frac{x_k^T z_k}{n}$ . We choose a value for the centering parameter  $\sigma_k \in (0, 1)$  and form the Jacobian of  $F_{\sigma_k, \mu_k}^{\text{IPM}}(\cdot)$ , evaluated at  $w_k$ . Then, the Newton direction is determined by solving the following system of equations:

$$\begin{bmatrix} -Q & A^T & I \\ A & 0 & 0 \\ Z_k & 0 & X_k \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta y_k \\ \Delta z_k \end{bmatrix} = - \begin{bmatrix} -c - Qx_k + A^T y_k + z_k \\ Ax_k - b \\ X_k Z_k e_n - \sigma_k \mu_k e_n \end{bmatrix}. \quad (6)$$

Notice that as  $\mu_k \rightarrow 0$ , the optimal solution of (3)-(4) converges to the optimal solution of (P)-(D). Polynomial convergence of such methods has been established multiple times in the literature (see for example [36, 37] and the references therein).

### 1.3 Proximal Point Methods

#### 1.3.1 Primal Proximal Point Method

One possible method for solving the primal problem (P), is the so called proximal point method. Given an arbitrary starting point  $x_0$ , the  $k$ -th iteration of the method is summarized by the following minimization problem:

$$x_{k+1} = \arg \min_x \left\{ c^T x + \frac{1}{2} x^T Q x + \frac{\mu_k}{2} \|x - x_k\|_2^2, \quad \text{s.t. } Ax = b, x \geq 0 \right\},$$

where  $\mu_k > 0$  is a non-increasing sequence of penalty parameters, and  $x_k$  is the current estimate for an optimal solution of (P) (the use of  $\mu_k$  is not a mistake; as will become clearer in Section 2, we intend to use the barrier parameter  $\mu_k$  to control both the IPM and the proximal method). Notice that such an algorithm is not practical, since we have to solve a sequence of sub-problems of similar difficulty to that of (P). Nevertheless, the proximal method contributes the  $\mu_k I$  term to the Hessian of the objective, and hence the sub-problems are strongly convex. This method is known to achieve a linear convergence rate (and possibly superlinear if  $\mu_k \rightarrow 0$  at a suitable rate), as shown in [13, 31], among others, even in the case where the minimization sub-problems are solved approximately. Various extensions of this method have been proposed in the literature. For example, one could employ different penalty functions other than the typical 2-norm penalty presented previously (e.g. see [9, 11, 19]).

Despite the fact that such algorithms are not practical, they have served as a powerful theoretical tool and a basis for other important methods. For an extensive review of the applications of standard primal proximal point methods, we refer the reader to [28].

#### 1.3.2 Dual Proximal Point Method

It is a well-known fact, observed for the first time in [31], that the application of the proximal point method on the dual problem, is equivalent to the application of the *augmented Lagrangian method* on the primal problem, which was proposed for the first time in [18, 30]. In view of the previous fact, we will present the derivation of the augmented Lagrangian method for the primal problem (P), while

having in mind that an equivalent reformulation of the model results in the proximal point method for the dual (D) (see [8, Chapter 3.4.4] or [12, 31]).

We start by defining the augmented Lagrangian function of (P). Given an arbitrary starting guess  $y_0$  for the dual variable, the augmented Lagrangian function is defined at the  $k$ -th iteration as:

$$\mathcal{L}_{\mu_k}^{\text{ALM}}(x; y_k) := c^T x + \frac{1}{2} x^T Q x - y_k^T (Ax - b) + \frac{1}{2\mu_k} \|Ax - b\|_2^2, \quad (7)$$

where  $\mu_k > 0$  is a non-increasing sequence of penalty parameters and  $y_k$  is the current estimate of an optimal Lagrange multiplier vector. The augmented Lagrangian method (ALM) is summarized below:

$$\begin{aligned} x_{k+1} &= \arg \min_x \{ \mathcal{L}_{\mu_k}^{\text{ALM}}(x; y_k), \text{ s.t. } x \geq 0 \}, \\ y_{k+1} &= y_k - \frac{1}{\mu_k} (Ax_{k+1} - b). \end{aligned}$$

Notice that the update of the Lagrange multiplier estimates can be interpreted as the application of the dual ascent method. A different type of update would be possible, however, the presented update is cheap and effective, due to the strong concavity of the proximal (“regularized”) dual problem (since  $\mu_k > 0$ ). Convergence and iteration complexity of such methods have been established multiple times (see for example [21, 31]). There is a vast literature on the subject of augmented Lagrangian methods. For a plethora of technical results and references, the reader is referred to [6]. For convergence results of various extended versions of the method, the reader is referred to [11].

It is worth noting here that a common issue, arising when using augmented Lagrangian methods, is that of the choice of the penalty parameter. To the authors’ knowledge, an optimal tuning for this parameter is still unknown.

### 1.3.3 Proximal Method of Multipliers

In [31], the author presented, for the first time, the *proximal method of multipliers* (PMM). The method consists of applying both primal and dual proximal point methods for solving (P)-(D). For an arbitrary starting point  $(x_0, y_0)$ , and using the already defined augmented Lagrangian function, given in (7), PMM can be summarized by the following iteration:

$$\begin{aligned} x_{k+1} &= \arg \min_x \{ \mathcal{L}_{\mu_k}^{\text{ALM}}(x; y_k) + \frac{\mu_k}{2} \|x - x_k\|_2^2, \text{ s.t. } x \geq 0 \}, \\ y_{k+1} &= y_k - \frac{1}{\mu_k} (Ax_{k+1} - b). \end{aligned} \quad (8)$$

One can observe that at every iteration of the method, the primal problem that we have to solve is strongly convex, while its dual, strongly concave. As shown in [31], the addition of the term  $\frac{\mu_k}{2} \|x - x_k\|_2^2$  in the augmented Lagrangian method does not affect its convergence rate, while ensuring better numerical behaviour of its iterates. An extension of this algorithm, allowing for the use of more general penalties, can be found in [11].

We can write (8) equivalently by making use of the maximal monotone operator  $T_{\mathcal{L}} : \mathbb{R}^{n+m} \rightrightarrows \mathbb{R}^{n+m}$  associated to (P)-(D) (see [31, 32]), that is defined as:

$$T_{\mathcal{L}}(x, y) := \{(u, v) : v \in Qx + c - A^T y + \partial \delta_+(x), u = Ax - b\}, \quad (9)$$

where  $\delta_+(\cdot)$  is an indicator function defined as:

$$\delta_+(x) := \begin{cases} \infty & \text{if } \exists j : x^j < 0, \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

and  $\partial(\cdot)$  denotes the sub-differential of a function. In our case, we have that (see [33, Section 23]):

$$z \in \partial \delta_+(x) \Leftrightarrow \begin{cases} z^j = 0 & \text{if } x^j > 0, \\ z^j \leq 0 & \text{if } x^j = 0, \end{cases}, \quad \forall j = \{1, \dots, n\}.$$

By convention, if there exists a component  $j$  such that  $x_j < 0$ , we have that  $\partial\delta_+(x) = \{\emptyset\}$ . Given a bounded pair  $(x^*, y^*)$  such that  $0 \in T_{\mathcal{L}}(x^*, y^*)$ , we can retrieve a vector  $z^* \in \partial\delta_+(x^*)$ , using which  $(x^*, y^*, -z^*)$  is an optimal solution for (P)-(D). By letting:

$$P_k := (I_{(m+n)} + \frac{1}{\mu_k} T_{\mathcal{L}})^{-1}, \quad (11)$$

we can express (8) as:

$$(x_{k+1}, y_{k+1}) = P_k(x_k, y_k), \quad (12)$$

and it can be shown that  $P_k$  is single valued and non-expansive (see [32]).

#### 1.4 Regularization in Interior Point Methods

In the context of interior point methods, it is often beneficial to include a regularization in order to improve the spectral properties of the system matrix in (6). For example, notice that if the constraint matrix  $A$  is rank-deficient, then the matrix in (6) might not be invertible. The latter can be immediately addressed by the introduction of a dual regularization, say  $\delta > 0$ , ensuring that  $\text{rank}([A \mid \delta I_m]) = m$ . For a detailed analysis of the effect of dual regularization on such systems, the reader is referred to [2]. On the other hand, the most common and efficient approach in practice is to eliminate the variables  $\Delta z$  from system (6), and solve the following symmetric reduced (*augmented*) system instead:

$$\begin{bmatrix} -(Q + \Theta_k^{-1}) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} = \begin{bmatrix} c + Qx_k - A^T y_k - \sigma_k \mu_k X_k^{-1} e \\ b - Ax_k \end{bmatrix}, \quad (13)$$

where  $\Theta_k = X_k Z_k^{-1}$ . Since  $X_k z_k \rightarrow 0$  close to optimality, one can observe that the matrix  $\Theta_k$  will contain some very large and some very small elements. Hence, the matrix in (13) will be increasingly ill-conditioned, as we approach optimality. The introduction of a primal regularization, say  $\rho > 0$ , can ensure that the matrix  $Q + \Theta_k^{-1} + \rho I_n$  will have eigenvalues that are bounded away from zero, and hence a significantly better worst-case conditioning than that of  $Q + \Theta_k^{-1}$ . In other words, regularization is commonly employed in IPMs, as a means of improving robustness and numerical stability (see [1]). As we will discuss later, by introducing regularization in IPMs, one can also gain efficiency. This is because regularization transforms the matrix in (13) into a quasi-definite one. It is known that such matrices can be factorized efficiently (see [34]).

In view of the previous discussion, one can observe that a very natural way of introducing primal regularization to problem (P), is through the application of the primal proximal point method. Similarly, dual regularization can be incorporated through the application of the dual proximal point method. This is a well-known fact. The authors in [1] presented a primal-dual regularized IPM, and interpreted this regularization as the application of the proximal point method. Consequently, the authors in [12] developed a primal-dual regularized IPM, which applies PMM to solve (P), and a single IPM iteration is employed for approximating the solution of each PMM sub-problem. There, global convergence of the method was proved, under some assumptions. A variation of the method proposed in [12] is given in [29], where general non-diagonal regularization matrices are employed, as a means of further improving factorization efficiency.

Similar ideas have been applied in the context of IPMs for general non-linear programming problems. For example, the authors in [3] presented an interior point method combined with the augmented Lagrangian method, and proved that, under some general assumptions, the method converges at a superlinear rate. Similar approaches can be found in [4, 5, 14], among others.

In this paper, we develop a path-following primal-dual regularized IPM for solving convex quadratic programming problems. The algorithm is obtained by applying one or a few iterations of the infeasible primal-dual interior point method in order to solve sub-problems arising from the primal-dual proximal point method. Under standard assumptions, we prove polynomial complexity of the algorithm and provide global convergence guarantees. To our knowledge, this is the first polynomial complexity result for a general primal-dual regularized IPM scheme. Notice that a complexity result is given for a primal regularized IPM for linear complementarity problems in [38]. However, the authors significantly alter the Newton directions, making their method very difficult to generalize and hard to achieve efficiency in practice. An important feature of the presented method is that it makes use of only one penalty parameter, that is the logarithmic penalty term. The aforementioned penalty has

been extensively studied and understood, and as a result, IPMs achieve fast and reliable convergence in practice. This is not the case for the penalty parameter of the proximal methods. In other words, IP-PMM inherits the fast and reliable convergence properties of IPMs, as well as the strong convexity of the PMM sub-problems, hence improving the conditioning of the Newton system solved at each IPM iteration, while providing a reliable tuning for the penalty parameter, independently of the problem at hand. The proposed approach is implemented and its reliability is demonstrated through extensive experimentation. The implementation slightly deviates from the theory, however, most of the theoretical results are verified in practice. The main purpose of this paper is to provide a reliable method that can be used for solving general convex quadratic problems, without the need of pre-processing, or of previous knowledge about the problems. The implemented method is supported by a novel theoretical result, indicating that regularization alleviates various issues arising in IPMs, without affecting their important worst-case polynomial complexity. As a side-product of the theory, an implementable infeasibility detection mechanism is also derived and tested in practice.

Before completing this section, let us introduce some notation that is used in the rest of this paper. An iteration of the algorithm is denoted by  $k \in \mathbb{N}$ . Given an arbitrary matrix (vector)  $A$  ( $x$ , respectively),  $A_k$  (or  $x_k$ ) denotes that  $A$  (resp.,  $x$ ) depends on the iteration  $k$ . An optimal solution to the pair (P)-(D) will be denoted as  $(x^*, y^*, z^*)$ . Optimal solutions of different primal-dual pairs will be denoted using an appropriate subscript, in order to distinguish them. For example, we use the notation  $(x_r^*, y_r^*, z_r^*)$ , for representing an optimal solution for a PMM sub-problem. The subscript is employed for distinguishing the “regularized” solution, from the solution of the initial problem, that is  $(x^*, y^*, z^*)$ . Any norm (semi-norm respectively) is denoted by  $\|\cdot\|_\nu$ , where  $\nu$  is used to distinguish between different norms. For example, the 2-norm (Euclidean norm) will be denoted as  $\|\cdot\|_2$ . When a given scalar is assumed to be independent of  $n$  and  $m$ , we mean that this quantity does not depend on the problem dimensions. Given two vectors of the same size  $x, y$ ,  $x \geq y$  denotes that the inequality holds component-wise. Given two logical statements  $T_1, T_2$ , the condition  $T_1 \wedge T_2$  is true only when both  $T_1$  and  $T_2$  are true. Let an arbitrary matrix  $A$  be given. The maximum (minimum) singular value of  $A$  is denoted as  $\eta_{\max}(A)$  ( $\eta_{\min}(A)$ ). Similarly, if  $A$  is square, its maximum eigenvalue is denoted as  $\nu_{\max}(A)$ . Given a set of indices, say  $\mathcal{B} \subset \{1, \dots, n\}$ , and an arbitrary vector (or matrix)  $x \in \mathbb{R}^n$  ( $A \in \mathbb{R}^{n \times n}$ ),  $x^{\mathcal{B}}$  ( $A^{\mathcal{B}}$ , respectively) denotes that sub-vector (sub-matrix) containing the elements of  $x$  (columns and rows of  $A$ ) whose indices belong to  $\mathcal{B}$ . Finally, the cardinality of  $\mathcal{B}$  is denoted as  $|\mathcal{B}|$ .

The rest of the paper is organized as follows. In Section 2, we provide the algorithmic framework of the method. Consequently, in Section 3, we prove polynomial complexity of the algorithm, and global convergence is established. In Section 4, a necessary condition for infeasibility is derived, which is later used to construct an infeasibility detection mechanism. Numerical results of the implemented method are presented and discussed in Section 5. Finally, we derive some conclusions in Section 6.

## 2 Algorithmic Framework

In the previous section, we presented all the necessary tools for deriving the proposed Interior Point-Proximal Method of Multipliers (IP-PMM). Effectively, we would like to merge the proximal method of multipliers with the infeasible interior point method. For that purpose, assume that, at some iteration  $k$  of the method, we have available an estimate  $\lambda_k$  for a Lagrange multiplier vector. Similarly, we denote by  $\zeta_k$  the estimate of a primal solution. We define the proximal penalty function that has to be minimized at the  $k$ -th iteration of PMM, for solving (P), given the estimates  $\lambda_k, \zeta_k$ , as:

$$\mathcal{L}_{\mu_k}^{PMM}(x; \zeta_k, \lambda_k) := c^T x + \frac{1}{2} x^T Q x - \lambda_k^T (Ax - b) + \frac{1}{2\mu_k} \|Ax - b\|_2^2 + \frac{\mu_k}{2} \|x - \zeta_k\|_2^2, \quad (14)$$

with  $\mu_k > 0$  some sequence of non-increasing penalty parameters. In order to solve the PMM sub-problem (8), we will apply one (or a few) iterations of the previously presented infeasible IPM. To do that, we alter (14), by including logarithmic barriers, that is:

$$\mathcal{L}_{\mu_k}^{IP-PMM}(x; \zeta_k, \lambda_k) := \mathcal{L}_{\mu_k}^{PMM}(x; \zeta_k, \lambda_k) - \mu_k \sum_{j=1}^n \ln x^j, \quad (15)$$

and we treat  $\mu_k$  as the barrier parameter. In order to form the optimality conditions of this sub-problem, we equate the gradient of  $\mathcal{L}_{\mu_k}^{IP-PMM}(\cdot; \lambda_k, \zeta_k)$  to the zero vector, i.e.:

$$c + Qx - A^T \lambda_k + \frac{1}{\mu_k} A^T (Ax - b) + \mu_k (x - \zeta_k) - \mu_k X^{-1} e_n = 0.$$



Following the developments in [3], we can define the variables  $y = \lambda_k - \frac{1}{\mu_k}(Ax - b)$  and  $z = \mu_k X^{-1}e_n$ , to get the following (equivalent) system of equations (first-order optimality conditions):

$$\begin{bmatrix} c + Qx - A^T y - z + \mu_k(x - \zeta_k) \\ Ax + \mu_k(y - \lambda_k) - b \\ Xz - \mu_k e_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (16)$$

Let us introduce the following notation, that will be used later.

**Definition 1** Let two real-valued positive functions be given:  $T(x) : \mathbb{R}_+ \mapsto \mathbb{R}_+$ ,  $f(x) : \mathbb{R}_+ \mapsto \mathbb{R}_+$ , with  $\mathbb{R}_+ := \{x \in \mathbb{R} : x \geq 0\}$ . We say that:

- $T(x) = O(f(x))$  (as  $x \rightarrow \infty$ ) if and only if, there exist constants  $c > 0$ ,  $x_0 \geq 0$ , such that:

$$T(x) \leq cf(x), \quad \text{for all } x \geq x_0.$$

- $T(x) = \Omega(f(x))$  if and only if, there exist constants  $c > 0$ ,  $x_0 \geq 0$ , such that:

$$T(x) \geq cf(x), \quad \text{for all } x \geq x_0.$$

- $T(x) = \Theta(f(x))$  if and only if,  $T(x) = O(f(x))$  and  $T(x) = \Omega(f(x))$ .

Next, given two arbitrary vectors  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}^n$ , we define the following semi-norm:

$$\|(b, c)\|_{\mathcal{A}} := \min_{x, y, z} \{ \|(x, z)\|_2 : Ax = b, -Qx + A^T y + z = c \}. \quad (17)$$

This semi-norm has been used before in [24], as a way to measure infeasibility for the case of linear programming problems ( $Q = 0$ ). For a discussion on the properties of the aforementioned semi-norm, as well as how one can evaluate it (using the QR factorization of  $A$ ), the reader is referred to [24, Section 4].

*Starting Point.* At this point, we are ready to define the starting point for IP-PMM. For that, we set  $(x_0, z_0) = \rho(e_n, e_n)$ , for some  $\rho > 0$ . We also set  $y_0$  to some arbitrary vector (e.g.  $y_0 = e_m$ ), such that  $\|y_0\|_\infty = O(1)$  (i.e. the absolute value of its entries is independent of  $n$  and  $m$ ), and  $\mu_0 = \frac{x_0^T z_0}{n}$ . Then we have:

$$Ax_0 = b + \bar{b}, \quad -Qx_0 + A^T y_0 + z_0 = c + \bar{c}, \quad \zeta_0 = x_0, \quad \lambda_0 = y_0. \quad (18)$$

for some vectors  $\bar{b}$ ,  $\bar{c}$ .

*Neighbourhood.* We mentioned earlier that we develop a path-following method. Hence, we have to describe a neighbourhood in which the iterations of the method should lie. However, unlike typical path-following methods, we define a family of neighbourhoods that depend on the PMM sub-problem parameters.

Given the starting point in (18), penalty  $\mu_k$ , and estimates  $\lambda_k, \zeta_k$ , we define the following *regularized set of centers*:

$$\mathcal{P}_{\mu_k}(\zeta_k, \lambda_k) := \{(x, y, z) \in \mathcal{C}_{\mu_k}(\zeta_k, \lambda_k) : (x, z) > (0, 0), Xz = \mu_k e_n\},$$

where

$$\mathcal{C}_{\mu_k}(\zeta_k, \lambda_k) := \left\{ (x, y, z) : \begin{array}{l} Ax + \mu_k(y - \lambda_k) = b + \frac{\mu_k}{\mu_0} \bar{b}, \\ -Qx + A^T y + z - \mu_k(x - \zeta_k) = c + \frac{\mu_k}{\mu_0} \bar{c} \end{array} \right\},$$

and  $\bar{b}$ ,  $\bar{c}$  are as in (18). The term set of centers originates from [24], where a similar set is studied.

In order to enlarge the previous set, we define the following set:

$$\tilde{\mathcal{C}}_{\mu_k}(\zeta_k, \lambda_k) := \left\{ (x, y, z) : \begin{array}{l} Ax + \mu_k(y - \lambda_k) = b + \frac{\mu_k}{\mu_0} (\bar{b} + \tilde{b}_k), \\ -Qx + A^T y + z - \mu_k(x - \zeta_k) = c + \frac{\mu_k}{\mu_0} (\bar{c} + \tilde{c}_k) \end{array} \right\},$$

$$\|(\tilde{b}_k, \tilde{c}_k)\|_2 \leq C_N, \quad \|(\tilde{b}_k, \tilde{c}_k)\|_{\mathcal{A}} \leq \gamma_{\mathcal{A}} \rho$$

where  $C_N > 0$  is a constant,  $\gamma_{\mathcal{A}} \in (0, 1)$ , and  $\rho > 0$  is as defined in the starting point. The vectors  $\tilde{b}_k$  and  $\tilde{c}_k$  represent the current scaled (by  $\frac{\mu_0}{\mu_k}$ ) infeasibility and vary depending on the iteration  $k$ . In particular, these vectors can be formally defined recursively, depending on the iterations of IP-PMM. However, such a definition is not necessary for the developments to follow. In essence, the



only requirement is that these scaled infeasibility vectors are bounded above by some constants, with respect to the 2-norm as well as the semi-norm defined in (17). Using the latter set, we are now ready to define a family of neighbourhoods:

$$\mathcal{N}_{\mu_k}(\zeta_k, \lambda_k) := \{(x, y, z) \in \tilde{\mathcal{C}}_{\mu_k}(\zeta_k, \lambda_k) : (x, z) > (0, 0), x^i z^i \geq \gamma_\mu \mu_k, i \in \{1, \dots, n\}\}, \quad (19)$$

where  $\gamma_\mu \in (0, 1)$  is a constant preventing component-wise complementarity products from approaching zero faster than  $\mu_k = \frac{x_k^T z_k}{n}$ . Obviously, the starting point defined in (18) belongs to the neighbourhood  $\mathcal{N}_{\mu_0}(\zeta_0, \lambda_0)$ , with  $(\bar{b}_0, \bar{c}_0) = (0, 0)$ . Notice from the definition of the neighbourhood, that it depends on the choice of the constants  $C_N, \gamma_A, \gamma_\mu$ . However, as the neighbourhood also depends on the parameters  $\mu_k, \lambda_k, \zeta_k$ , we omit the dependence on the constants, for simplicity of notation.

*Newton System.* At every IP-PMM iteration, we approximately solve a perturbed form of the conditions in (16), by applying a variation of Newton's method. In particular, we form the Jacobian of the left-hand side of (16) and we perturb the right-hand side of the Newton equation as follows:

$$\begin{bmatrix} -(Q + \mu_k I_n) & A^T & I \\ A & \mu_k I_m & 0 \\ Z_k & 0 & X_k \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \Delta y_k \\ \Delta z_k \end{bmatrix} = - \begin{bmatrix} -(c + \frac{\sigma_k \mu_k}{\mu_0} \bar{c}) - Qx_k + A^T y_k + z_k - \sigma_k \mu_k (x_k - \zeta_k) \\ Ax_k + \sigma_k \mu_k (y_k - \lambda_k) - (b + \frac{\sigma_k \mu_k}{\mu_0} \bar{b}) \\ X_k Z_k e_n - \sigma_k \mu_k e_n \end{bmatrix}, \quad (20)$$

where  $\bar{b}, \bar{c}$  are as in (18). Notice that we perturb the right-hand side of the Newton system in order to ensure that the iterates remain in the neighbourhood (19), while trying to reduce the value of the penalty (barrier) parameter  $\mu_k$ .

We are now able to derive Algorithm [IP-PMM](#), summarizing the proposed interior point-proximal method of multipliers. We will prove polynomial complexity of this scheme in the next section, under standard assumptions.

---

**Algorithm IP-PMM** Interior Point-Proximal Method of Multipliers

---

**Input:**  $A, Q, b, c, \text{tol}$ .

**Parameters:**  $0 < \sigma_{\min} \leq \sigma_{\max} \leq 0.5, C_N > 0, 0 < \gamma_A < 1, 0 < \gamma_\mu < 1$ .

**Starting point:** Set as in (18).

```

for ( $k = 0, 1, 2, \dots$ ) do
  if ( $(\|Ax_k - b\|_2 < \text{tol}) \wedge (\|c + Qx_k - A^T y_k - z_k\|_2 < \text{tol}) \wedge (\mu_k < \text{tol})$ ) then
    return  $(x_k, y_k, z_k)$ .
  else
    Choose  $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$  and solve (20).
    Choose step-length  $\alpha_k$ , as the largest  $\alpha \in (0, 1]$  such that  $\mu_k(\alpha) \leq (1 - 0.01\alpha)\mu_k$  and:

```

$$(x_k + \alpha_k \Delta x_k, y_k + \alpha_k \Delta y_k, z_k + \alpha_k \Delta z_k) \in \mathcal{N}_{\mu_k(\alpha)}(\zeta_k, \lambda_k),$$

$$\text{where, } \mu_k(\alpha) = \frac{(x_k + \alpha_k \Delta x_k)^T (z_k + \alpha_k \Delta z_k)}{n}.$$

Set  $(x_{k+1}, y_{k+1}, z_{k+1}) = (x_k + \alpha_k \Delta x_k, y_k + \alpha_k \Delta y_k, z_k + \alpha_k \Delta z_k)$ ,  $\mu_{k+1} = \frac{x_{k+1}^T z_{k+1}}{n}$ .

Let  $r_p = Ax_{k+1} - (b + \frac{\mu_{k+1}}{\mu_0} \bar{b})$ ,  $r_d = (c + \frac{\mu_{k+1}}{\mu_0} \bar{c}) + Qx_{k+1} - A^T y_{k+1} - z_{k+1}$ .

**if**  $\left( (\|r_p, r_d\|_2 \leq C_N \frac{\mu_{k+1}}{\mu_0}) \wedge (\|(r_p, r_d)\|_A \leq \gamma_A \rho \frac{\mu_{k+1}}{\mu_0}) \right)$  **then**

$(\zeta_{k+1}, \lambda_{k+1}) = (x_{k+1}, y_{k+1})$ .

**else**

$(\zeta_{k+1}, \lambda_{k+1}) = (\zeta_k, \lambda_k)$ .

**end if**

**end for**

---

Notice, in Algorithm [IP-PMM](#), that we force  $\sigma$  to be less than 0.5. This value is set, without loss of generality, for simplicity of exposition. Similarly, in the choice of the step-length, we require that  $\mu_k(\alpha) \leq (1 - 0.01\alpha)\mu_k$ . The constant 0.01 is chosen for ease of presentation. It depends on the choice of

the maximum value of  $\sigma$ . The constants  $C_N$ ,  $\gamma_A$ ,  $\gamma_\mu$ , are used in the definition of the neighbourhood in (19). Their values can be considered to be arbitrary. The input  $\text{tol}$ , represents the error tolerance (chosen by the user). The terminating conditions require the Euclidean norm of primal and dual infeasibility, as well as complementarity, to be less than this tolerance. In such a case, we accept the iterate as a solution triple. The estimates  $\lambda$ ,  $\zeta$  are not updated if primal or dual infeasibility are not both sufficiently decreased. In this case, we keep the estimates constant while continuing decreasing the penalty parameter  $\mu_k$ . Following the usual practice with proximal and augmented Lagrangian methods, we accept a new estimate when the respective residual is sufficiently decreased. However, the algorithm requires the evaluation of the semi-norm defined in (17), at every iteration. While this is not practical, it can be achieved in polynomial time, with respect to the size of the problem. For a detailed discussion on this, the reader is referred to [24, Section 4].

### 3 Convergence Analysis of IP-PMM

In this section we prove polynomial complexity and global convergence of Algorithm IP-PMM. The proof follows by induction on the iterations of IP-PMM. That is, given an iterate  $(x_k, y_k, z_k)$  at an arbitrary iteration  $k$ , we prove that the next iterate belongs to the appropriate neighbourhood required by the algorithm. In turn, this allows us to prove global and polynomial convergence of IP-PMM. An outline of the proof can be briefly explained as follows:

- Initially, we present some technical results in Lemmas 1-3 which are required for the analysis throughout this section.
- In turn, we prove boundedness of the iterates  $(x_k, y_k, z_k)$  of IP-PMM in Lemma 4. In particular we show that  $\|(x_k, y_k, z_k)\|_2 = O(n)$  and  $\|(x_k, z_k)\|_1 = O(n)$  for every  $k \geq 0$ .
- Then, we prove boundedness of the Newton direction computed at every IP-PMM iteration in Lemma 5. More specifically, we prove that  $\|(\Delta x_k, \Delta y_k, \Delta z_k)\|_2 = O(n^3)$  for every  $k \geq 0$ .
- In Lemma 6 we prove the existence of a positive step-length  $\bar{\alpha}$  so that the new iterate of IP-PMM,  $(x_{k+1}, y_{k+1}, z_{k+1})$ , belongs to the updated neighbourhood  $\mathcal{N}_{\mu_{k+1}}(\zeta_{k+1}, \lambda_{k+1})$ , for every  $k \geq 0$ . In particular, we show that  $\bar{\alpha} \geq \frac{\bar{\kappa}}{n^4}$ , where  $\bar{\kappa}$  is a constant independent of  $n$  and  $m$ .
- Q-linear convergence of the barrier parameter  $\mu_k$  (to zero) is established in Theorem 1.
- The polynomial complexity of IP-PMM is then proved in Theorem 2, showing that it converges to an  $\epsilon$ -accurate solution in at most  $O(n^4 |\log(\frac{1}{\epsilon})|)$  steps.
- Finally, global converge to an optimal solution of (P)-(D) is established in Theorem 3.

For the rest of this section, we will make use of the following two assumptions, which are commonly employed when analyzing the complexity of an IPM.

**Assumption 1** *There exists an optimal solution  $(x^*, y^*, z^*)$  for the primal-dual pair (P)-(D), such that  $\|x^*\|_\infty \leq C^*$ ,  $\|y^*\|_\infty \leq C^*$  and  $\|z^*\|_\infty \leq C^*$ , for some constant  $C^* \geq 0$ , independent of  $n$  and  $m$ .*

**Assumption 2** *The constraint matrix of (P) has full row rank, that is  $\text{rank}(A) = m$ . Furthermore, we assume that there exist constants  $C_A^1 > 0$ ,  $C_A^2 > 0$ ,  $C_Q > 0$  and  $C_r > 0$ , independent of  $n$  and  $m$ , such that:*

$$\eta_{\min}(A) \geq C_A^1, \quad \eta_{\max}(A) \leq C_A^2, \quad \nu_{\max}(Q) \leq C_Q, \quad \|(c, b)\|_\infty \leq C_r.$$

Note that the independence of the previous constants from the problem dimensions is assumed for simplicity of exposition; this is a common practice when analyzing the complexity of interior point methods. If these constants depend polynomially on  $n$  (or  $m$ ), the analysis still holds by suitably altering the worst-case polynomial bound for the number of iterations of the algorithm.

Let us now use the properties of the proximal operator defined in (11).

**Lemma 1** *Given Assumption 1, and for all  $\lambda \in \mathbb{R}^m$ ,  $\zeta \in \mathbb{R}^n$  and  $0 \leq \mu < \infty$ , there exists a unique pair  $(x_r^*, y_r^*)$ , such that  $(x_r^*, y_r^*) = P(\zeta, \lambda)$ , and*

$$\|(x_r^*, y_r^*) - (x^*, y^*)\|_2 \leq \|(\zeta, \lambda) - (x^*, y^*)\|_2, \quad (21)$$

where  $P(\cdot)$  is defined as in (11) and  $(x^*, y^*)$  is the same as in Assumption 1.

*Proof* We know that  $P(\cdot, \cdot)$  is single-valued and non-expansive ([32]), and hence there exists a unique pair  $(x_r^*, y_r^*)$ , such that  $(x_r^*, y_r^*) = P(\zeta, \lambda)$ , for all  $\lambda, \zeta$  and  $0 \leq \mu < \infty$ . Given the optimal triple of Assumption 1, we can use the non-expansiveness of  $P(\cdot)$  in (11), to show that:

$$\|P(\zeta, \lambda) - P(x^*, y^*)\|_2 = \|(x_r^*, y_r^*) - (x^*, y^*)\|_2 \leq \|(\zeta, \lambda) - (x^*, y^*)\|_2,$$

where we used the fact that  $P(x^*, y^*) = (x^*, y^*)$ , which follows directly from (9), as we can see that  $(0, 0) \in T_{\mathcal{L}}(x^*, y^*)$ . This completes the proof.  $\square$

**Lemma 2** *Given Assumptions 1, 2, there exists a triple  $(x_{r_k}^*, y_{r_k}^*, z_{r_k}^*)$ , satisfying:*

$$\begin{aligned} Ax_{r_k}^* + \mu(y_{r_k}^* - \lambda_k) - b &= 0 \\ -c - Qx_{r_k}^* + A^T y_{r_k}^* + z_{r_k}^* - \mu(x_{r_k}^* - \zeta_k) &= 0, \\ (x_{r_k}^*)^T (z_{r_k}^*) &= 0, \end{aligned} \quad (22)$$

with  $\|(x_{r_k}^*, y_{r_k}^*, z_{r_k}^*)\|_2 = O(\sqrt{n})$ , for all  $\lambda_k \in \mathbb{R}^m$ ,  $\zeta_k \in \mathbb{R}^n$  produced by Algorithm IP-PMM, and any  $\mu \in [0, \infty)$ . Moreover, we have that  $\|(\zeta_k, \lambda_k)\|_2 = O(\sqrt{n})$ , for all  $k \geq 0$ .

*Proof* We prove the claim by induction on the iterates,  $k \geq 0$ , of Algorithm IP-PMM. At iteration  $k = 0$ , we have that  $\lambda_0 = y_0$  and  $\zeta_0 = x_0$ . But from the construction of the starting point in (18), we know that  $\|(x_0, y_0)\|_2 = O(\sqrt{n})$ . Hence,  $\|(\zeta_0, \lambda_0)\|_2 = O(\sqrt{n})$  (assuming that  $n > m$ ). From Lemma 1, we know that there exists a unique pair  $(x_{r_0}^*, y_{r_0}^*)$  such that:

$$(x_{r_0}^*, y_{r_0}^*) = P_0(\zeta_0, \lambda_0), \quad \text{and} \quad \|(x_{r_0}^*, y_{r_0}^*) - (x^*, y^*)\|_2 \leq \|(\zeta_0, \lambda_0) - (x^*, y^*)\|_2.$$

Using the triangular inequality, and combining the latter inequality with our previous observations, as well as Assumption 1, yields that  $\|(x_{r_0}^*, y_{r_0}^*)\|_2 = O(\sqrt{n})$ . From the definition of the operator in (12), we know that:

$$\begin{aligned} -c - Qx_{r_0}^* + A^T y_{r_0}^* - \mu(x_{r_0}^* - \zeta_k) &\in \partial\delta_+(x_{r_0}^*), \\ Ax_{r_0}^* + \mu(y_{r_0}^* - \lambda_k) - b &= 0, \end{aligned}$$

where  $\partial\delta_+(\cdot)$  is the sub-differential of the indicator function defined in (10). Hence, we know that there must exist  $-z_{r_0}^* \in \partial\delta_+(x_{r_0}^*)$  (and hence,  $z_{r_0}^* \geq 0$ ,  $(x_{r_0}^*)^T (z_{r_0}^*) = 0$ ), such that:

$$z_{r_0}^* = c + Qx_{r_0}^* - A^T y_{r_0}^* + \mu(x_{r_0}^* - \zeta_k), \quad (x_{r_0}^*)^T (z_{r_0}^*) = 0, \quad \|z_{r_0}^*\|_2 = O(\sqrt{n}),$$

where  $\|z_{r_0}^*\|_2 = O(\sqrt{n})$  follows from Assumption 2, combined with  $\|(x_0, y_0)\|_2 = O(\sqrt{n})$ .

Let us now assume that at some iteration  $k$  of Algorithm IP-PMM, we have  $\|(\zeta_k, \lambda_k)\|_2 = O(\sqrt{n})$ . There are two cases for the subsequent iteration:

1. The proximal estimates are updated, that is  $(\zeta_{k+1}, \lambda_{k+1}) = (x_{k+1}, y_{k+1})$ , or
2. the proximal estimates stay the same, i.e.  $(\zeta_{k+1}, \lambda_{k+1}) = (\zeta_k, \lambda_k)$ .

**Case 1.** We know by construction that this case can only occur if the following condition is satisfied:

$$\|(r_p, r_d)\|_2 \leq C_N \frac{\mu_{k+1}}{\mu_0},$$

where  $r_p, r_d$  are defined in Algorithm IP-PMM. However, from the neighbourhood conditions in (19), we know that:

$$\|(r_p + \mu_{k+1}(y_{k+1} - \lambda_k), r_d + \mu_{k+1}(x_{k+1} - \zeta_k))\|_2 \leq C_N \frac{\mu_{k+1}}{\mu_0}.$$

Combining the last two inequalities by applying the triangular inequality, and using the inductive hypothesis ( $\|(\lambda_k, \zeta_k)\|_2 = O(\sqrt{n})$ ), yields that  $\|(x_{k+1}, y_{k+1})\|_2 = O(\sqrt{n})$ . Hence,  $\|(\zeta_{k+1}, \lambda_{k+1})\|_2 = O(\sqrt{n})$ . Then, we can invoke Lemma 1, with  $\lambda = \lambda_{k+1}$ ,  $\zeta = \zeta_{k+1}$  and  $\mu = \mu_{k+1}$ , which gives that:

$$\|(x_{r_{k+1}}^*, y_{r_{k+1}}^*) - (x^*, y^*)\|_2 \leq \|(\zeta_{k+1}, \lambda_{k+1}) - (x^*, y^*)\|_2.$$

A simple manipulation yields that  $\|(x_{r_{k+1}}^*, y_{r_{k+1}}^*)\|_2 = O(\sqrt{n})$ . As before, we use (12) alongside Assumption 2 to show the existence of  $-z_{r_{k+1}}^* \in \partial\delta_+(x_{r_{k+1}}^*)$ , such that the triple  $(x_{r_{k+1}}^*, y_{r_{k+1}}^*, z_{r_{k+1}}^*)$  satisfies (22) with  $\|z_{r_{k+1}}^*\|_2 = O(\sqrt{n})$ .

**Case 2.** In this case, we have  $(\zeta_{k+1}, \lambda_{k+1}) = (\zeta_k, \lambda_k)$ , and hence the inductive hypothesis gives us directly that:  $\|(\zeta_{k+1}, \lambda_{k+1})\|_2 = O(\sqrt{n})$ . The same reasoning as before implies the existence of a triple  $(x_{r_{k+1}}^*, y_{r_{k+1}}^*, z_{r_{k+1}}^*)$  satisfying (22), with  $\|(x_{r_{k+1}}^*, y_{r_{k+1}}^*, z_{r_{k+1}}^*)\|_2 = O(\sqrt{n})$ .  $\square$

**Lemma 3** *Given Assumptions 1, 2,  $\lambda_k$  and  $\zeta_k$  produced at an arbitrary iteration  $k \geq 0$  of Algorithm IP-PMM and any  $\mu \in [0, \infty)$ , there exists a triple  $(\tilde{x}, \tilde{y}, \tilde{z})$  which satisfies the following system of equations:*

$$\begin{aligned} A\tilde{x} + \mu\tilde{y} &= b + \bar{b} + \mu\lambda_k + \tilde{b}_k, \\ -Q\tilde{x} + A^T\tilde{y} + \tilde{z} - \mu\tilde{x} &= c + \bar{c} - \mu\zeta_k + \tilde{c}_k, \\ \tilde{X}\tilde{z} &= \theta e_n, \end{aligned} \quad (23)$$

for some arbitrary  $\theta > 0$  ( $\theta = \Theta(1)$ ), with  $(\tilde{x}, \tilde{z}) \geq \xi(e_n, e_n)$  ( $\xi = \Theta(1)$ ) and  $\|(\tilde{x}, \tilde{y}, \tilde{z})\|_2 = O(\sqrt{n})$ , where  $\tilde{b}_k, \tilde{c}_k$  are defined in (19), while  $\bar{b}, \bar{c}$  are defined with the starting point in (18).

*Proof* Let  $k \geq 0$  denote an arbitrary iteration of Algorithm IP-PMM. Let also  $\bar{b}, \bar{c}$  as defined in (18), and  $\tilde{b}_k, \tilde{c}_k$ , as defined in the neighbourhood conditions in (19). Given an arbitrary positive constant  $\theta > 0$ , we consider the following barrier primal-dual pair:

$$\min_x \left( (c + \bar{c} + \tilde{c}_k)^T x + \frac{1}{2} x^T Q x - \theta \sum_{j=1}^n \ln x^j \right), \quad \text{s.t. } Ax = b + \bar{b} + \tilde{b}_k, \quad (24)$$

$$\max_{x,y,z} \left( (b + \bar{b} + \tilde{b}_k)^T y - \frac{1}{2} x^T Q x + \theta \sum_{j=1}^n \ln x^j \right), \quad \text{s.t. } -Qx + A^T y + z = c + \bar{c} + \tilde{c}_k. \quad (25)$$

Let us now define the following triple:

$$(\hat{x}, \hat{y}, \hat{z}) := \arg \min_{(x,y,z)} \{ \|(x, z)\|_2 : Ax = \tilde{b}_k, -Qx + A^T y + z = \tilde{c}_k \}.$$

From the neighbourhood conditions (19), we know that  $\|(\tilde{b}_k, \tilde{c}_k)\|_A \leq \gamma_A \rho$ , and from the definition of the semi-norm in (17), we have that:  $\|(\hat{x}, \hat{z})\|_2 \leq \gamma_A \rho$ . Using (17) alongside Assumption 2, we can also show that  $\|\hat{y}\|_2 = \Theta(\|(\hat{x}, \hat{z})\|_2)$ . On the other hand, from the definition of the starting point, we have that:  $(x_0, z_0) = \rho(e_n, e_n)$ . By defining the following auxiliary point:

$$(\bar{x}, \bar{y}, \bar{z}) = (x_0, y_0, z_0) + (\hat{x}, \hat{y}, \hat{z}),$$

we have that  $(\bar{x}, \bar{z}) \geq (1 - \gamma_A)\rho(e_n, e_n)$ . By construction, the triple  $(\bar{x}, \bar{y}, \bar{z})$  is a feasible solution for the primal-dual pair in (24)-(25), giving bounded primal and dual objective values, respectively.

Using our previous observations, alongside the fact that  $\text{rank}(A) = m$  (Assumption 2), we can confirm that there must exist a large constant  $M > 0$ , and a triple  $(x_s^*, y_s^*, z_s^*)$  solving (24)-(25), such that  $\|(x_s^*, z_s^*)\|_\infty \leq M \Rightarrow \|(x_s^*, y_s^*, z_s^*)\|_2 = O(\sqrt{n})$ .

Let us now apply the proximal method of multipliers to (24)-(25), given the estimates  $\zeta_k, \lambda_k$ . We should note at this point, that the proximal operator used here is different from that in (11), since it is based on a different maximal monotone operator from that in (9). In particular, we associate the following maximal monotone operator to (24)-(25):

$$\tilde{T}_{\mathcal{L}}(x, y) := \{(u, v) : v = Qx + (c + \bar{c} + \tilde{c}_k) - A^T y - \theta X^{-1} e_n, u = Ax - (b + \bar{b} + \tilde{b}_k)\},$$

As before, the proximal operator is defined as  $\tilde{P} := (I_{m+n} + \tilde{T}_{\mathcal{L}})^{-1}$ , and is single-valued and non-expansive. We let any  $\mu \in [0, \infty)$  and define the following penalty function:

$$\begin{aligned} \tilde{\mathcal{L}}_{\mu, \theta}(x; \zeta_k, \lambda_k) &:= (c + \bar{c} + \tilde{c}_k)^T x + \frac{1}{2} x^T Q x + \\ &\frac{1}{2\mu} \|x - \zeta_k\|_2^2 + \frac{1}{2\mu} \|Ax - (b + \bar{b} + \tilde{b}_k)\|_2^2 - (\lambda_k)^T (Ax - (b + \bar{b} + \tilde{b}_k)) - \theta \sum_{j=1}^n \ln x^j. \end{aligned}$$

By defining the variables  $y = \lambda_k - \frac{1}{\mu}(Ax - (b + \bar{b} + \tilde{b}_k))$  and  $z = \theta X^{-1} e_n$ , we can see that the optimality conditions of this PMM sub-problem are exactly those stated in (23). Equivalently, we can find a pair  $(\tilde{x}, \tilde{y})$  such that  $(\tilde{x}, \tilde{y}) = \tilde{P}(\zeta_k, \lambda_k)$  and set  $\tilde{z} = \theta \tilde{X}^{-1} e_n$ . Then, we can use the non-expansiveness of  $\tilde{P}$ , as in Lemma 1, to obtain:

$$\|(\tilde{x}, \tilde{y}) - (x_s^*, y_s^*)\|_2 \leq \|(\zeta_k, \lambda_k) - (x_s^*, y_s^*)\|_2.$$

But we know, from Lemma 2, that  $\|(\zeta_k, \lambda_k)\|_2 = O(\sqrt{n})$ , for all  $k \geq 0$ . Combining this with our previous observations yields that  $\|(\tilde{x}, \tilde{y})\|_2 = O(\sqrt{n})$ . Setting  $\tilde{z} = \theta \tilde{X}^{-1} e_n$ , gives a triple  $(\tilde{x}, \tilde{y}, \tilde{z})$

that satisfies (23), while  $\|(\tilde{x}, \tilde{y}, \tilde{z})\|_2 = O(\sqrt{n})$ . To conclude the proof, let us notice that the value of  $\tilde{\mathcal{L}}_{\mu, \theta}(x; \zeta_k, \lambda_k)$  will grow unbounded as  $x \rightarrow 0$  or  $x \rightarrow \infty$ . Hence, there must exist a constant  $\tilde{M} > 0$ , such that the minimizer of this function must satisfy  $\frac{1}{\tilde{M}} \leq \tilde{x}^j \leq \tilde{M}$ , for all  $j \in \{1, \dots, n\}$ . The relation  $\tilde{X}\tilde{z} = \theta e_n$  then implies that  $\frac{\theta}{\tilde{M}} \leq \tilde{z}^j \leq \theta \tilde{M}$ . Hence  $(\tilde{x}, \tilde{z}) \geq \xi(e_n, e_n)$ , where  $\xi > 0$  and  $\xi = \Theta(1)$ .  $\square$

In the following Lemma, we derive boundedness of the iterates of Algorithm IP-PM.

**Lemma 4** *Given Assumptions 1 and 2, the iterates  $(x_k, y_k, z_k)$  produced by Algorithm IP-PM, for all  $k \geq 0$ , are such that:*

$$\|(x_k, z_k)\|_1 = O(n), \quad \|(x_k, y_k, z_k)\|_2 = O(n).$$

*Proof* Let an iterate  $(x_k, y_k, z_k) \in \mathcal{N}_{\mu_k}(\zeta_k, \lambda_k)$ , produced by Algorithm IP-PM during an arbitrary iteration  $k \geq 0$ , be given. Firstly, we invoke Lemma 3, from which we have a triple  $(\tilde{x}, \tilde{y}, \tilde{z})$  satisfying (23), for  $\mu = \mu_k$ . Similarly, by invoking Lemma 2, we know that there exists a triple  $(x_{r_k}^*, y_{r_k}^*, z_{r_k}^*)$  satisfying (22), with  $\mu = \mu_k$ . Consider the following auxiliary point:

$$\left( \left(1 - \frac{\mu_k}{\mu_0}\right)x_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{x} - x_k, \left(1 - \frac{\mu_k}{\mu_0}\right)y_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{y} - y_k, \left(1 - \frac{\mu_k}{\mu_0}\right)z_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{z} - z_k \right). \quad (26)$$

Using (26) and (22)-(23) (for  $\mu = \mu_k$ ), one can observe that:

$$\begin{aligned} A\left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{x} - x_k\right) + \mu_k\left(\left(1 - \frac{\mu_k}{\mu_0}\right)y_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{y} - y_k\right) &= \\ \left(1 - \frac{\mu_k}{\mu_0}\right)(Ax_{r_k}^* + \mu_k y_{r_k}^*) + \frac{\mu_k}{\mu_0}(A\tilde{x} + \mu_k \tilde{y}) - Ax_k - \mu_k y_k &= \\ \left(1 - \frac{\mu_k}{\mu_0}\right)(b + \mu_k \lambda_k) + \frac{\mu_k}{\mu_0}(b + \mu_k \lambda_k + \tilde{b} + \bar{b}) - Ax_k - \mu_k y_k &= \\ b + \mu_k \lambda_k + \frac{\mu_k}{\mu_0}(\tilde{b} + \bar{b}) - Ax_k - \mu_k y_k &= 0, \end{aligned}$$

where the last equality follows from the definition of the neighbourhood  $\mathcal{N}_{\mu_k}(\zeta_k, \lambda_k)$ . Similarly:

$$-(Q + \mu_k I_n)\left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{x} - x_k\right) + A^T\left(\left(1 - \frac{\mu_k}{\mu_0}\right)y_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{y} - y_k\right) + \left(\left(1 - \frac{\mu_k}{\mu_0}\right)z_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{z} - z_k\right) = 0.$$

By combining the previous two relations, we have:

$$\begin{aligned} &\left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{x} - x_k\right)^T \left(\left(1 - \frac{\mu_k}{\mu_0}\right)z_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{z} - z_k\right) = \\ &\left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{x} - x_k\right)^T (Q + \mu_k I)\left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{x} - x_k\right) + \\ &\mu_k\left(\left(1 - \frac{\mu_k}{\mu_0}\right)y_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{y} - y_k\right)^T \left(\left(1 - \frac{\mu_k}{\mu_0}\right)y_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{y} - y_k\right) \geq 0. \end{aligned} \quad (27)$$

From (27), it can be seen that:

$$\begin{aligned} &\left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{x}\right)^T z_k + \left(\left(1 - \frac{\mu_k}{\mu_0}\right)z_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{z}\right)^T x_k \leq \\ &\left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{x}\right)^T \left(\left(1 - \frac{\mu_k}{\mu_0}\right)z_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{z}\right) + x_k^T z_k. \end{aligned}$$

However, from Lemmas 2 and 3, we have that:  $(\tilde{x}, \tilde{z}) \geq \xi(e_n, e_n)$ , for some positive constant  $\xi = \Theta(1)$ , while  $\|(x_{r_k}^*, z_{r_k}^*)\|_2 = O(\sqrt{n})$ , and,  $\|(\tilde{x}, \tilde{z})\|_2 = O(\sqrt{n})$ . Furthermore, by definition we have that  $n\mu_k = x_k^T z_k$ . By combining all the previous, we obtain:

$$\begin{aligned} &\frac{\mu_k}{\mu_0}\xi(e^T x_k + e^T z_k) \leq \\ &\left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{x}\right)^T z_k + \left(\left(1 - \frac{\mu_k}{\mu_0}\right)z_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{z}\right)^T x_k \leq \\ &\left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{x}\right)^T \left(\left(1 - \frac{\mu_k}{\mu_0}\right)z_{r_k}^* + \frac{\mu_k}{\mu_0}\tilde{z}\right) + x_k^T z_k = \\ &\frac{\mu_k}{\mu_0}\left(1 - \frac{\mu_k}{\mu_0}\right)(x_{r_k}^*)^T \tilde{z} + \frac{\mu_k}{\mu_0}\left(1 - \frac{\mu_k}{\mu_0}\right)\tilde{x}^T z_{r_k}^* + \left(\frac{\mu_k}{\mu_0}\right)^2 \tilde{x}^T \tilde{z} + x_k^T z_k = O(\mu_k n), \end{aligned} \quad (28)$$

where we used (22) ( $(x_{r_k}^*)^T (z_{r_k}^*) = 0$ ). Hence, (28) implies that:

$$\|(x_k, z_k)\|_1 = O(n).$$

From equivalence of norms, we have that  $\|(x_k, z_k)\|_2 \leq \|(x_k, z_k)\|_1$ . Finally, from the neighbourhood conditions we know that:

$$c + Qx_k - A^T y_k - z_k + \mu_k(x_k - \zeta_k) + \frac{\mu_k}{\mu_0}(\tilde{c}_k + \bar{c}) = 0.$$

All terms above (except for  $y_k$ ) have a 2-norm that is  $O(n)$  (note that  $\|(\bar{c}, \tilde{b})\|_2 = O(\sqrt{n})$  using Assumption 2 and the definition in (18)). Hence, using again Assumption 2 yields that  $\|y_k\|_2 = O(n)$ , and completes the proof.  $\square$

As in a typical IPM convergence analysis, we proceed by bounding some components of the scaled Newton direction. The proof of that uses similar arguments to those presented in [36, Lemma 6.5]. Combining this result with Assumption 2, allows us to bound also the unscaled Newton direction.

**Lemma 5** *Given Assumptions 1 and 2, and the Newton direction  $(\Delta x_k, \Delta y_k, \Delta z_k)$  obtained by solving system (20) during an arbitrary iteration  $k \geq 0$  of Algorithm IP-PMM, we have that:*

$$\|D_k^{-1} \Delta x_k\|_2 = O(n^2 \mu^{\frac{1}{2}}), \quad \|D_k \Delta z_k\|_2 = O(n^2 \mu^{\frac{1}{2}}), \quad \|(\Delta x_k, \Delta y_k, \Delta z_k)\|_2 = O(n^3),$$

with  $D_k^2 := X_k Z_k^{-1}$ .

*Proof* Consider an arbitrary iteration  $k$  of Algorithm IP-PMM. We invoke Lemmas 2, 3, for  $\mu = \sigma_k \mu_k$ . That is, there exists a triple  $(x_{r_k}^*, y_{r_k}^*, z_{r_k}^*)$  satisfying (22), and a triple  $(\tilde{x}, \tilde{y}, \tilde{z})$  satisfying (23), for  $\mu = \sigma_k \mu_k$ . Using the centering parameter  $\sigma_k$ , we define the following vectors:

$$\hat{c} := -\left(\sigma_k \frac{\bar{c}}{\mu_0} - (1 - \sigma_k)(x_k - \zeta_k + \frac{\mu_k}{\mu_0}(\tilde{x} - x_{r_k}^*))\right), \quad \hat{b} := -\left(\sigma_k \frac{\bar{b}}{\mu_0} + (1 - \sigma_k)(y_k - \lambda_k + \frac{\mu_k}{\mu_0}(\tilde{y} - y_{r_k}^*))\right), \quad (29)$$

where  $\bar{b}$ ,  $\bar{c}$ ,  $\mu_0$  are defined in (18). Using Lemmas 2, 3, 4, and Assumption 2, we know that  $\|(\hat{c}, \hat{b})\|_2 = O(n)$ . Then, by applying again Assumption 2, we know that there must exist a vector  $\hat{x}$  such that:  $A\hat{x} = \hat{b}$ ,  $\|\hat{x}\|_2 = O(n)$ , and by setting  $\hat{z} = \hat{c} + Q\hat{x} + \mu\hat{x}$ , we have that  $\|\hat{z}\|_2 = O(n)$  and:

$$A\hat{x} = \hat{b}, \quad -Q\hat{x} + \hat{z} - \mu\hat{x} = \hat{c}. \quad (30)$$

Using  $(x_{r_k}^*, y_{r_k}^*, z_{r_k}^*)$ ,  $(\tilde{x}, \tilde{y}, \tilde{z})$ , as well as the triple  $(\hat{x}, 0, \hat{z})$ , where  $(\hat{x}, \hat{z})$  is defined in (30), we can define the following auxiliary triple:

$$(\bar{x}, \bar{y}, \bar{z}) = (\Delta x_k, \Delta y_k, \Delta z_k) + \frac{\mu_k}{\mu_0}(\tilde{x}, \tilde{y}, \tilde{z}) - \frac{\mu_k}{\mu_0}(x_{r_k}^*, y_{r_k}^*, z_{r_k}^*) + \mu_k(\hat{x}, 0, \hat{z}). \quad (31)$$

Using (31), (29), (22)-(23) (with  $\mu = \sigma_k \mu_k$ ), and the second block equation of (20), we have:

$$\begin{aligned} A\bar{x} + \mu_k \bar{y} &= (A\Delta x_k + \mu_k \Delta y_k) + \frac{\mu_k}{\mu_0}((A\tilde{x} + \mu_k \tilde{y}) - (Ax_{r_k}^* + \mu_k y_{r_k}^*)) + \mu_k A\hat{x} \\ &= (b + \sigma_k \frac{\mu_k}{\mu_0} \bar{b} - Ax_k - \sigma_k \mu_k (y_k - \lambda_k)) + \frac{\mu_k}{\mu_0}((A\tilde{x} + \mu_k \tilde{y}) - (Ax_{r_k}^* + \mu_k y_{r_k}^*)) \\ &\quad - \mu_k \left(\sigma_k \frac{\bar{b}}{\mu_0} + (1 - \sigma_k)(y_k - \lambda_k)\right) - \frac{\mu_k}{\mu_0}(1 - \sigma_k) \mu_k (\tilde{y} - y_{r_k}^*) \\ &= (b + \sigma_k \frac{\mu_k}{\mu_0} \bar{b} - Ax_k - \sigma_k \mu_k (y_k - \lambda_k)) + \frac{\mu_k}{\mu_0}(b + \sigma_k \mu_k \lambda_k + \bar{b} + \tilde{b}_k) \\ &\quad - \frac{\mu_k}{\mu_0}(\sigma_k \mu_k \lambda_k + b) - \mu_k \left(\sigma_k \frac{\bar{b}}{\mu_0} + (1 - \sigma_k)(y_k - \lambda_k)\right) \\ &= b + \frac{\mu_k}{\mu_0}(\bar{b} + \tilde{b}_k) - Ax_k - \mu_k (y_k - \lambda_k) \\ &= 0, \end{aligned}$$

where the last equation follows from the neighbourhood conditions (i.e.  $(x_k, y_k, z_k) \in \mathcal{N}_{\mu_k}(\zeta_k, \lambda_k)$ ). Similarly, we can show that:

$$-Q\bar{x} + A^T \bar{y} + \bar{z} - \mu_k \bar{x} = 0.$$

The previous two equalities imply that:

$$\bar{x}^T \bar{z} = \bar{x}^T (Q\bar{x} - A^T \bar{y} + \mu_k \bar{x}) = \bar{x}^T (Q + \mu_k I) \bar{x} + \mu_k \bar{y}^T \bar{y} \geq 0. \quad (32)$$

On the other hand, using the last block equation of the Newton system (20), we have:

$$Z_k \bar{x} + X_k \bar{z} = -X_k Z_k e_n + \sigma_k \mu_k e_n + \frac{\mu_k}{\mu_0} Z_k (\tilde{x} - x_{r_k}^*) + \frac{\mu_k}{\mu_0} X_k (\tilde{z} - z_{r_k}^*) + \mu_k Z_k \hat{x} + \mu_k X_k \hat{z}.$$

Let  $W_k = (X_k Z_k)^{\frac{1}{2}}$ . By multiplying both sides of the previous equation by  $W_k^{-1}$ , we get:

$$D_k^{-1} \bar{x} + D_k \bar{z} = -W_k^{-1} (X_k Z_k e_n - \sigma_k \mu_k e_n) + \frac{\mu_k}{\mu_0} (D_k^{-1} (\tilde{x} - x_{r_k}^*) + D_k (\tilde{z} - z_{r_k}^*)) + \mu_k (D_k^{-1} \hat{x} + D_k \hat{z}). \quad (33)$$

But, from (32), we know that  $\bar{x}^T \bar{z} \geq 0$  and hence:

$$\|D_k^{-1} \bar{x} + D_k \bar{z}\|_2^2 \geq \|D_k^{-1} \bar{x}\|_2^2 + \|D_k \bar{z}\|_2^2.$$

Combining (33) with the previous inequality, gives:

$$\begin{aligned} \|D_k^{-1} \bar{x}\|_2^2 + \|D_k \bar{z}\|_2^2 &\leq \left\{ \|W_k^{-1}\|_2 \|X_k Z_k e_n - \sigma_k \mu_k e_n\|_2 + \right. \\ &\quad \left. \frac{\mu_k}{\mu_0} (\|D_k^{-1} (\tilde{x} - x_{r_k}^*)\|_2 + \|D_k (\tilde{z} - z_{r_k}^*)\|_2) + \mu_k (\|D_k^{-1} \hat{x}\|_2 + \|D_k \hat{z}\|_2) \right\}^2. \end{aligned}$$

We isolate one of the two terms of the left hand side of the previous inequality, take square roots on both sides, use (31) and apply the triangular inequality to it, to obtain:

$$\begin{aligned} \|D_k^{-1} \Delta x_k\|_2 &\leq \|W_k^{-1}\|_2 \|X_k Z_k e_n - \sigma_k \mu_k e_n\|_2 \\ &\quad + \frac{\mu_k}{\mu_0} (2\|D_k^{-1} (\tilde{x} - x_{r_k}^*)\|_2 + \|D_k (\tilde{z} - z_{r_k}^*)\|_2) + \mu_k (2\|D_k^{-1} \hat{x}\|_2 + \|D_k \hat{z}\|_2). \end{aligned} \quad (34)$$

We now proceed to bounding the terms in the right hand side of (34). Firstly, notice from the neighbourhood conditions (19) that  $\gamma_\mu \mu_k \leq x_k^i z_k^i$ . This in turn implies that:

$$\|W_k^{-1}\|_2 = \max_i \frac{1}{(x_k^i z_k^i)^{\frac{1}{2}}} \leq \frac{1}{(\gamma_\mu \mu_k)^{\frac{1}{2}}}.$$

On the other hand, we have that:

$$\begin{aligned} \|X_k Z_k e_n - \sigma_k \mu_k e_n\|_2^2 &= \|X_k Z_k e_n\|_2^2 - 2\sigma_k \mu_k x_k^T z_k + \sigma_k^2 \mu_k^2 n \\ &\leq \|X_k Z_k e_n\|_1^2 - 2\sigma_k \mu_k x_k^T z_k + \sigma_k^2 \mu_k^2 n \\ &= (\mu_k n)^2 - 2\sigma_k \mu_k^2 n + \sigma_k \mu_k^2 n \\ &\leq \mu_k^2 n^2. \end{aligned}$$

Hence, combining the previous two relations yields:

$$\|W_k^{-1}\|_2 \|X_k Z_k e_n - \sigma_k \mu_k e_n\|_2 \leq \frac{n}{\gamma_\mu^{\frac{1}{2}} \mu_k^{\frac{1}{2}}} = O(n \mu_k^{\frac{1}{2}}).$$

We proceed by bounding  $\|D_k^{-1}\|_2$ . For that, using Lemma 4, we have:

$$\|D_k^{-1}\|_2 = \max_i |(D_k^{ii})^{-1}| = \|D_k^{-1} e_n\|_\infty = \|W_k^{-1} Z_k e_n\|_\infty \leq \|W_k^{-1}\|_2 \|z_k\|_1 = O\left(\frac{n}{\mu_k^{\frac{1}{2}}}\right).$$

Similarly, we have that:

$$\|D_k\|_2 = O\left(\frac{n}{\mu_k^{\frac{1}{2}}}\right).$$

Hence, using the previous bounds, as well as Lemmas 2, 3, we obtain:

$$\begin{aligned} 2 \frac{\mu_k}{\mu_0} \|D_k^{-1} (\tilde{x} - x_{r_k}^*)\|_2 + \frac{\mu_k}{\mu_0} \|D_k (\tilde{z} - z_{r_k}^*)\|_2 &\leq 2 \frac{\mu_k}{\mu_0} (\|D_k^{-1}\|_2 + \|D_k\|_2) \max\{\|\tilde{x} - x_{r_k}^*\|_2, \|\tilde{z} - z_{r_k}^*\|_2\} \\ &= O(n^{\frac{3}{2}} \mu_k^{\frac{1}{2}}), \end{aligned}$$

and

$$\mu_k (2\|D_k^{-1} \hat{x}\|_2 + \|D_k \hat{z}\|_2) \leq 2\mu_k (\|D_k^{-1}\|_2 + \|D_k\|_2) \max\{\|\hat{x}\|_2, \|\hat{z}\|_2\} = O(n^2 \mu_k^{\frac{1}{2}}).$$



Combining all the previous bounds yields the claimed bound for  $\|D_k^{-1}\Delta x_k\|_2$ . One can bound  $\|D_k\Delta z_k\|_2$  in the same way. The latter is omitted for ease of presentation.

Finally, we have that:

$$\|\Delta x_k\|_2 = \|D_k D_k^{-1} \Delta x_k\|_2 \leq \|D_k\|_2 \|D_k^{-1} \Delta x_k\|_2 = O(n^3).$$

Similarly, we can show that  $\|\Delta z_k\|_2 = O(n^3)$ . From the first block equation of the Newton system in (20), alongside Assumption 2, we can show that  $\|\Delta y_k\|_2 = O(n^3)$ , which completes the proof.  $\square$

We are now able to prove that at every iteration of Algorithm IP-PMM, there exists a step-length  $\alpha_k > 0$ , using which, the new iterate satisfies the conditions required by the algorithm.

**Lemma 6** *Given Assumptions 1 and 2, there exists a step-length  $\bar{\alpha} \in (0, 1)$ , such that for all  $\alpha \in [0, \bar{\alpha}]$  and for all iterations  $k \geq 0$  of Algorithm IP-PMM, the following relations hold:*

$$(x_k + \alpha \Delta x_k)^T (z_k + \alpha \Delta z_k) \geq (1 - \alpha(1 - \beta_1)) x_k^T z_k, \quad (35)$$

$$(x_k^i + \alpha \Delta x_k^i)(z_k^i + \alpha \Delta z_k^i) \geq \frac{\gamma \mu}{n} (x_k + \alpha \Delta x_k)^T (z_k + \alpha \Delta z_k), \text{ for all } i \in \{1, \dots, n\}, \quad (36)$$

$$(x_k + \alpha \Delta x_k)^T (z_k + \alpha \Delta z_k) \leq (1 - \alpha(1 - \beta_2)) x_k^T z_k, \quad (37)$$

where, without loss of generality,  $\beta_1 = \frac{\sigma_{\min}}{2}$  and  $\beta_2 = 0.99$ . Moreover,  $\bar{\alpha} \geq \frac{\bar{\kappa}}{n^4}$  for all  $k \geq 0$ , where  $\bar{\kappa} > 0$  is independent of  $n$  and  $m$ , and if  $(x_k, y_k, z_k) \in \mathcal{N}_{\mu_k}(\zeta_k, \lambda_k)$ , then letting:

$$(x_{k+1}, y_{k+1}, z_{k+1}) = (x_k + \alpha \Delta x_k, y_k + \alpha \Delta y_k, z_k + \alpha \Delta z_k), \quad \mu_{k+1} = \frac{x_{k+1}^T z_{k+1}}{n}, \text{ for all } \alpha \in (0, \bar{\alpha}]$$

gives  $(x_{k+1}, y_{k+1}, z_{k+1}) \in \mathcal{N}_{\mu_{k+1}}(\zeta_{k+1}, \lambda_{k+1})$ , where  $\lambda_k, \zeta_k$  are updated as in Algorithm IP-PMM.

*Proof* In order to prove the first three inequalities, we follow the developments in [36, Chapter 6, Lemma 6.7]. From Lemma 5, we have that there exists a constant  $C_\Delta > 0$ , such that:

$$(\Delta x_k)^T \Delta z_k = (D_k^{-1} \Delta x_k)^T (D_k \Delta z_k) \leq \|D_k^{-1} \Delta x_k\|_2 \|D_k \Delta z_k\|_2 \leq C_\Delta^2 n^4 \mu_k.$$

Similarly, it is easy to see that:

$$|\Delta x_k^i \Delta z_k^i| \leq C_\Delta^2 n^4 \mu_k.$$

On the other hand, by summing over all  $n$  components of the last block equation of the Newton system (20), we have:

$$z_k^T \Delta x_k + x_k^T \Delta z_k = e_n^T (Z_k \Delta x_k + X_k \Delta z_k) = e_n^T (-X_k Z_k e_n + \sigma_k \mu_k e_n) = (\sigma_k - 1) x_k^T z_k, \quad (38)$$

while the components of the last block equation of the Newton system (20) can be written as:

$$z_k^i \Delta x_k^i + x_k^i \Delta z_k^i = -x_k^i z_k^i + \sigma_k \mu_k. \quad (39)$$

We proceed by proving (35). Using (38), we have:

$$\begin{aligned} (x_k + \alpha \Delta x_k)^T (z_k + \alpha \Delta z_k) - (1 - \alpha(1 - \beta_1)) x_k^T z_k &= \\ x_k^T z_k + \alpha(\sigma_k - 1) x_k^T z_k + \alpha^2 \Delta x_k^T \Delta z_k - (1 - \alpha) x_k^T z_k - \alpha \beta_1 x_k^T z_k &\geq \\ \alpha(\sigma_k - \beta_1) x_k^T z_k - \alpha^2 C_\Delta^2 n^4 \mu_k &\geq \alpha \left( \frac{\sigma_{\min}}{2} \right) n \mu_k - \alpha^2 C_\Delta^2 n^4 \mu_k, \end{aligned}$$

where we set (without loss of generality)  $\beta_1 = \frac{\sigma_{\min}}{2}$ . The most-right hand side of the previous inequality will be non-negative for every  $\alpha$  satisfying:

$$\alpha \leq \frac{\sigma_{\min}}{2 C_\Delta^2 n^3}.$$

In order to prove (36), we will use (39) and the fact that from the neighbourhood conditions we have that  $x_k^i z_k^i \geq \gamma \mu_k$ . In particular, we obtain:

$$\begin{aligned} (x_k^i + \alpha \Delta x_k^i)(z_k^i + \alpha \Delta z_k^i) &\geq (1 - \alpha) x_k^i z_k^i + \alpha \sigma_k \mu_k - \alpha^2 C_\Delta^2 n^4 \mu_k \\ &\geq \gamma \mu (1 - \alpha) \mu_k + \alpha \sigma_k \mu_k - \alpha^2 C_\Delta^2 n^4 \mu_k. \end{aligned}$$

By combining all the previous, we get:

$$\begin{aligned} (x_k^i + \alpha \Delta x_k^i)(z_k^i + \alpha \Delta z_k^i) - \frac{\gamma_\mu}{n}(x_k + \alpha \Delta x_k)^T(z_k + \alpha \Delta z_k) &\geq \\ \alpha \sigma_k(1 - \gamma_\mu)\mu_k - (1 + \frac{\gamma_\mu}{n})\alpha^2 C_\Delta^2 n^4 \mu_k &\geq \\ \alpha \sigma_{\min}(1 - \gamma_\mu)\mu_k - 2\alpha^2 C_\Delta^2 n^4 \mu_k. \end{aligned}$$

In turn, the most-right hand side of the previous relation will be non-negative for every  $\alpha$  satisfying:

$$\alpha \leq \frac{\sigma_{\min}(1 - \gamma_\mu)}{2C_\Delta^2 n^4}.$$

Finally, to prove (37), we set (without loss of generality)  $\beta_2 = 0.99$ . We know, from Algorithm IP-PMM, that  $\sigma_{\max} \leq 0.5$ . With the previous two remarks in mind, we have:

$$\begin{aligned} \frac{1}{n}(x_k + \alpha \Delta x_k)^T(z_k + \alpha \Delta z_k) - (1 - 0.01\alpha)\mu_k &\leq \\ (1 - \alpha)\mu_k + \alpha \sigma_k \mu_k + \alpha^2 \frac{C_\Delta^2 n^4}{n} \mu_k - (1 - 0.01\alpha)\mu_k &\leq \\ -0.99\alpha \mu_k + 0.5\alpha \mu_k + \alpha^2 \frac{C_\Delta^2 n^4}{n} \mu_k &= \\ -0.49\alpha \mu_k + \alpha^2 \frac{C_\Delta^2 n^4}{n} \mu_k. \end{aligned}$$

The last term will be non-positive for every  $\alpha$  satisfying:

$$\alpha \leq \frac{0.49}{C_\Delta^2 n^3}.$$

By combining all the previous bounds on the step-length, we have that (35)-(37) will hold for every  $\alpha \in (0, \alpha^*)$ , where:

$$\alpha^* := \min \left\{ \frac{\sigma_{\min}}{2C_\Delta^2 n^3}, \frac{\sigma_{\min}(1 - \gamma_\mu)}{2C_\Delta^2 n^4}, \frac{0.49}{C_\Delta^2 n^3}, 1 \right\}. \quad (40)$$

Next, we would like to find the maximum  $\bar{\alpha} \in (0, \alpha^*]$ , such that:

$$(x_k(\alpha), y_k(\alpha), z_k(\alpha)) \in \mathcal{N}_{\mu_k(\alpha)}(\zeta_k, \lambda_k), \text{ for all } \alpha \in (0, \bar{\alpha}],$$

where  $\mu_k(\alpha) = \frac{x_k(\alpha)^T z_k(\alpha)}{n}$  and:

$$(x_k(\alpha), y_k(\alpha), z_k(\alpha)) = (x_k + \alpha \Delta x_k, y_k + \alpha \Delta y_k, z_k + \alpha \Delta z_k).$$

Let:

$$\tilde{r}_p(\alpha) = Ax_k(\alpha) + \mu_k(\alpha)(y_k(\alpha) - \lambda_k) - \left(b + \frac{\mu_k(\alpha)}{\mu_0}\bar{b}\right),$$

and

$$\tilde{r}_d(\alpha) = -Qx_k(\alpha) + A^T y_k(\alpha) + z_k(\alpha) - \mu_k(\alpha)(x_k(\alpha) - \zeta_k) - \left(c + \frac{\mu_k(\alpha)}{\mu_0}\bar{c}\right).$$

In other words, we need to find the maximum  $\bar{\alpha} \in (0, \alpha^*]$ , such that:

$$\|\tilde{r}_p(\alpha), \tilde{r}_d(\alpha)\|_2 \leq C_N \frac{\mu_k(\alpha)}{\mu_0}, \quad \|\tilde{r}_p(\alpha), \tilde{r}_d(\alpha)\|_A \leq \gamma_A \rho \frac{\mu_k(\alpha)}{\mu_0}, \text{ for all } \alpha \in (0, \bar{\alpha}]. \quad (41)$$

If the latter two conditions hold, then  $(x_k(\alpha), y_k(\alpha), z_k(\alpha)) \in \mathcal{N}_{\mu_k(\alpha)}(\zeta_k, \lambda_k)$ , for all  $\alpha \in (0, \bar{\alpha}]$ . Then, if Algorithm IP-PMM updates  $\zeta_k, \lambda_k$ , it does so only when similar conditions (as in (41)) hold for the new parameters. If the parameters are not updated, then the new iterate lies in the desired neighbourhood because of (41), alongside (35)-(37).

We start by rearranging  $\tilde{r}_p(\alpha)$ . Specifically, we have that:

$$\begin{aligned}\tilde{r}_p(\alpha) &= A(x_k + \alpha \Delta x_k) + (\mu_k + \alpha(\sigma_k - 1)\mu_k + \alpha^2 \frac{\Delta x_k^T \Delta z_k}{n})((y_k + \alpha \Delta y_k - \lambda_k) - \frac{\bar{b}}{\mu_0}) - b \\ &= (Ax_k + \mu_k(y_k - \lambda_k) - b - \frac{\mu_k \bar{b}}{\mu_0}) + \alpha(A\Delta x_k + \mu_k \Delta y_k) + \\ &\quad + (\alpha(\sigma_k - 1)\mu_k + \alpha^2 \frac{\Delta x_k^T \Delta z_k}{n})((y_k - \lambda_k + \alpha \Delta y_k) - \frac{\bar{b}}{\mu_0}) \\ &= \frac{\mu_k}{\mu_0} \tilde{b}_k + \alpha \left( b - Ax_k - \sigma_k \mu_k ((y_k - \lambda_k) - \frac{\bar{b}}{\mu_0}) + \mu_k ((y_k - \lambda_k) - \frac{\bar{b}}{\mu_0}) - \right. \\ &\quad \left. - \mu_k ((y_k - \lambda_k) - \frac{\bar{b}}{\mu_0}) \right) + (\alpha(\sigma_k - 1)\mu_k + \alpha^2 \frac{\Delta x_k^T \Delta z_k}{n})((y_k - \lambda_k + \alpha \Delta y_k) - \frac{\bar{b}}{\mu_0}),\end{aligned}$$

where we used that  $\mu_k(\alpha) = (\mu_k + \alpha(\sigma_k - 1)\mu_k + \alpha^2 \frac{\Delta x_k^T \Delta z_k}{n})$ , which can be derived from (38), as well as the neighbourhood conditions (19), and the second block equation of the Newton system (20). By using again the neighbourhood conditions, and then by deleting the opposite terms in the previous equation, we obtain:

$$\tilde{r}_p(\alpha) = (1 - \alpha) \frac{\mu_k}{\mu_0} \tilde{b}_k + \alpha^2 (\sigma_k - 1) \mu_k \Delta y_k + \alpha^2 \frac{\Delta x_k^T \Delta z_k}{n} (y_k - \lambda_k + \alpha \Delta y_k - \frac{\bar{b}}{\mu_0}). \quad (42)$$

Similarly, we can show that:

$$\tilde{r}_d(\alpha) = (1 - \alpha) \frac{\mu_k}{\mu_0} \tilde{c}_k - \alpha^2 (\sigma_k - 1) \mu_k \Delta x_k - \alpha^2 \frac{\Delta x_k^T \Delta z_k}{n} (x_k - \zeta_k + \alpha \Delta x_k + \frac{\bar{c}}{\mu_0}). \quad (43)$$

Define the following two quantities:

$$\begin{aligned}\xi_2 &:= \mu_k \|(\Delta y_k, \Delta x_k)\|_2 + C_\Delta^2 n^3 \mu_k \left( \|(y_k - \lambda_k, x_k - \zeta_k)\|_2 + \alpha^* \|(\Delta y_k, \Delta x_k)\|_2 + \left\| \left( \frac{\bar{b}}{\mu_0}, \frac{\bar{c}}{\mu_0} \right) \right\|_2 \right), \\ \xi_{\mathcal{A}} &:= \mu_k \|(\Delta y_k, \Delta x_k)\|_{\mathcal{A}} + C_\Delta^2 n^3 \mu_k \left( \|(y_k - \lambda_k, x_k - \zeta_k)\|_{\mathcal{A}} + \alpha^* \|(\Delta y_k, \Delta x_k)\|_{\mathcal{A}} + \left\| \left( \frac{\bar{b}}{\mu_0}, \frac{\bar{c}}{\mu_0} \right) \right\|_{\mathcal{A}} \right),\end{aligned} \quad (44)$$

where  $\alpha^*$  is given by (40). Using the definition of the starting point in (18), as well as results in Lemmas 4, 5, we can observe that  $\xi_2 = O(n^4 \mu_k)$ . On the other hand, using Assumption 2, we know that for every pair  $(r_1, r_2) \in \mathbb{R}^{m+n}$ , if  $\|(r_1, r_2)\|_2 = \Theta(f(n))$ , where  $f(\cdot)$  is a positive polynomial function of  $n$ , then  $\|(r_1, r_2)\|_{\mathcal{A}} = \Theta(f(n))$ . In other words, we have that  $\xi_{\mathcal{A}} = O(n^4 \mu_k)$ . Using the quantities in (44), equations (42), (43), as well as the neighbourhood conditions, we have that:

$$\begin{aligned}\|\tilde{r}_p(\alpha), \tilde{r}_d(\alpha)\|_2 &\leq (1 - \alpha) C_N \frac{\mu_k}{\mu_0} + \alpha^2 \mu_k \xi_2, \\ \|\tilde{r}_p(\alpha), \tilde{r}_d(\alpha)\|_{\mathcal{A}} &\leq (1 - \alpha) \gamma_{\mathcal{A}} \rho \frac{\mu_k}{\mu_0} + \alpha^2 \mu_k \xi_{\mathcal{A}},\end{aligned}$$

for all  $\alpha \in (0, \alpha^*]$ , where  $\alpha^*$  is given by (40). On the other hand, we know from (35), that:

$$\mu_k(\alpha) \geq (1 - \alpha(1 - \beta_1))\mu_k, \text{ for all } \alpha \in (0, \alpha^*].$$

By combining the last two inequalities, we get that:

$$\|\tilde{r}_p(\alpha), \tilde{r}_d(\alpha)\|_2 \leq \frac{\mu_k(\alpha)}{\mu_0} C_N, \quad \text{for all } \alpha \in \left(0, \min \left\{ \alpha^*, \frac{\beta_1 C_N}{\xi_2 \mu_0} \right\} \right].$$

Similarly,

$$\|\tilde{r}_p(\alpha), \tilde{r}_d(\alpha)\|_{\mathcal{A}} \leq \frac{\mu_k(\alpha)}{\mu_0} \gamma_{\mathcal{A}} \rho, \quad \text{for all } \alpha \in \left(0, \min \left\{ \alpha^*, \frac{\beta_1 \gamma_{\mathcal{A}} \rho}{\xi_{\mathcal{A}} \mu_0} \right\} \right].$$

Hence, we have that:

$$\bar{\alpha} := \min \left\{ \alpha^*, \frac{\beta_1 C_N}{\xi_2 \mu_0}, \frac{\beta_1 \gamma_{\mathcal{A}} \rho}{\xi_{\mathcal{A}} \mu_0} \right\}, \quad (45)$$

where  $\beta_1 = \frac{\sigma_{\min}}{2}$ . Since  $\bar{\alpha} = \Omega(\frac{1}{n^4})$ , we know that there must exist a constant  $\bar{\kappa} > 0$ , independent of  $n$ ,  $m$  and of the iteration  $k$ , such that  $\bar{\alpha} \geq \frac{\bar{\kappa}}{n^4}$  for all  $k \geq 0$ , and this completes the proof.  $\square$

The following Theorem summarizes our results.

**Theorem 1** *Given Assumptions 1, 2, the sequence  $\{\mu_k\}$  generated by Algorithm IP-PMM converges  $Q$ -linearly to zero, and the sequences of regularized residual norms*

$$\left\{ \left\| Ax_k + \mu_k(y_k - \lambda_k) - b - \frac{\mu_k}{\mu_0} \bar{b} \right\|_2 \right\} \text{ and } \left\{ \left\| -Qx_k + A^T y_k + z_k - \mu_k(x_k - \zeta_k) - c - \frac{\mu_k}{\mu_0} \bar{c} \right\|_2 \right\}$$

*converge  $R$ -linearly to zero.*

*Proof* From (37) we have that:

$$\mu_{k+1} \leq (1 - 0.01\alpha_k)\mu_k,$$

while, from (45), we know that for all  $k \geq 0$ , there exists  $\bar{\alpha} \geq \frac{\bar{\kappa}}{n^4}$  such that  $\alpha_k \geq \bar{\alpha}$ . Hence, we can easily see that  $\mu_k \rightarrow 0$ . On the other hand, from the neighbourhood conditions, we know that for all  $k \geq 0$ :

$$\left\| Ax_k + \mu_k(y_k - \lambda_k) - b - \frac{\mu_k}{\mu_0} \bar{b} \right\|_2 \leq C_N \frac{\mu_k}{\mu_0}$$

and

$$\left\| -Qx_k + A^T y_k + z_k - \mu_k(x_k - \zeta_k) - c - \frac{\mu_k}{\mu_0} \bar{c} \right\|_2 \leq C_N \frac{\mu_k}{\mu_0}.$$

This completes the proof.  $\square$

**Theorem 2** *Let  $\epsilon \in (0, 1)$  be a given error tolerance. Choose a starting point for Algorithm IP-PMM as in (18), such that  $\mu_0 \leq \frac{C}{\epsilon^\omega}$  for some positive constants  $C, \omega$ . Given Assumptions 1 and 2, there exists an index  $K$  with:*

$$K = O(n^4 |\log(\frac{1}{\epsilon})|)$$

*such that the iterates  $\{w_k\} = \{(x_k^T, y_k^T, z_k^T)^T\}$  generated from Algorithm IP-PMM satisfy:*

$$\mu_k \leq \epsilon, \quad \text{for all } k \geq K.$$

*Proof* The proof follows the developments in [36, 37] and is only provided here for completeness. Without loss of generality, we can chose  $\sigma_{\max} \leq 0.5$  and then from Lemma 6, we know that there is a constant  $\bar{\kappa}$  independent of  $n$  such that:  $\bar{a} \geq \frac{\bar{\kappa}}{n^4}$ , where  $\bar{a}$  is the worst-case step-length. Given the latter, we know that the new iterate lies in the neighbourhood  $\mathcal{N}_{\mu_{k+1}}(\zeta_{k+1}, \lambda_{k+1})$  defined in (19). We also know, from (37), that:

$$\mu_{k+1} \leq (1 - 0.01\bar{a})\mu_k \leq (1 - 0.01\frac{\bar{\kappa}}{n^4})\mu_k, \quad k = 0, 1, 2, \dots$$

By taking logarithms on both sides in the previous inequality, we get:

$$\log(\mu_{k+1}) \leq \log(1 - \frac{\tilde{\kappa}}{n^4}) + \log(\mu_k),$$

where  $\tilde{\kappa} = 0.01\bar{\kappa}$ . By applying repeatedly the previous formula, and using the fact that  $\mu_0 \leq \frac{C}{\epsilon^\omega}$ , we have:

$$\log(\mu_k) \leq k \log(1 - \frac{\tilde{\kappa}}{n^4}) + \log(\mu_0) \leq k \log(1 - \frac{\tilde{\kappa}}{n^4}) + \omega \log(\frac{1}{\epsilon}) + \log(C).$$

We use the fact that  $\log(1 + \beta) \leq \beta$ , for all  $\beta > -1$  to obtain:

$$\log(\mu_k) \leq k(-\frac{\tilde{\kappa}}{n^4}) + \omega \log(\frac{1}{\epsilon}) + \log(C).$$

Hence, convergence is attained if:

$$k(-\frac{\tilde{\kappa}}{n^4}) + \omega \log(\frac{1}{\epsilon}) + \log(C) \leq \log(\epsilon).$$

The latter holds for all  $k$  satisfying:

$$k \geq K = \frac{n^4}{\tilde{\kappa}} ((1 + \omega) \log(\frac{1}{\epsilon}) + \log(C)),$$

which completes the proof.  $\square$

Finally, we present the global convergence guarantee of Algorithm IP-PMM.

**Theorem 3** Suppose that Algorithm IP-PMM terminates when a limit point is reached. Then, if Assumptions 1 and 2 hold, every limit point of  $\{(x_k, y_k, z_k)\}$  determines a primal-dual solution of the non-regularized pair (P)-(D).

*Proof* From Theorem 1, we know that  $\{\mu_k\} \rightarrow 0$ , and hence, there exists a sub-sequence  $\mathcal{K} \subseteq \mathbb{N}$ , such that:

$$\{Ax_k + \mu_k(y_k - \lambda_k) - b - \frac{\mu_k}{\mu_0}\bar{b}\}_{\mathcal{K}} \rightarrow 0, \quad \{-Qx_k + A^T y_k + z_k - \mu_k(x_k - \zeta_k) - c - \frac{\mu_k}{\mu_0}\bar{c}\}_{\mathcal{K}} \rightarrow 0.$$

However, since Assumptions 1 and 2 hold, we know from Lemma 4 that  $\{(x_k, y_k, z_k)\}$  is a bounded sequence. Hence, we obtain that:

$$\{Ax_k - b\}_{\mathcal{K}} \rightarrow 0, \quad \{-Qx_k + A^T y_k + z_k - c\}_{\mathcal{K}} \rightarrow 0.$$

One can readily observe that the limit point of the algorithm satisfies the conditions given in (2), since  $\mu_k = \frac{x_k^T z_k}{n}$ .  $\square$

*Remark 1* Let us notice that the above analysis required Assumption 2, and hence that  $\text{rank}(A) = m$ . However, one motivation for using regularization is that the resulting algorithm is able to solve rank deficient problems. While the complexity result would not hold in this case, we are able to perform an analysis following exactly the developments in [12], which would guarantee the global convergence of IP-PMM, given certain assumptions (see [12]). The latter is omitted for brevity of presentation.

#### 4 Infeasible Problems

Let us now drop Assumptions 1, 2, in order to analyze the behaviour of the algorithm in the case where an infeasible problem is tackled. Let us employ the following two premises:

**Premise 1** During the iterations of Algorithm IP-PMM, the sequences  $\{\|y_k - \lambda_k\|_2\}$  and  $\{\|x_k - \zeta_k\|_2\}$ , remain bounded.

**Premise 2** There does not exist a primal-dual triple, satisfying the KKT conditions for the primal-dual pair (P)-(D).

The following analysis is based on the developments in [2] and [12]. However, in these papers such an analysis is proposed in order to derive convergence of an IPM, while here, we use it as a tool in order to construct a reliable and implementable infeasibility detection mechanism. In what follows, we show that Premises 1 and 2 are contradictory. In other words, if Premise 2 holds (which means that our problem is infeasible), then we will show that Premise 1 cannot hold, and hence the negation of Premise 1 is a necessary condition for infeasibility.

**Lemma 7** Given Premise 1, and by assuming that  $x_k^T z_k > \epsilon$ , for some  $\epsilon > 0$ , for all iterations  $k$  of Algorithm IP-PMM, the Newton direction produced by (20) is uniformly bounded by a constant dependent only on  $n$ .

*Proof* Let us use a variation of Theorem 1 given in [2]. This theorem states that if the following conditions are satisfied,

1.  $\mu_k > 0$ ,
2. there exists  $\bar{\epsilon} > 0$  :  $x_k^i z_k^i \geq \bar{\epsilon}$ , for all  $i = \{1, 2, \dots, n\}$ , and for all  $k \geq 0$ ,
3. and the matrix  $H_k = \mu_k I + Q + X_k^{-1} Z_k + \frac{1}{\mu_k} A^T A$  is positive definite,

then the Jacobian matrix in (20) is non-singular and has a uniformly bounded inverse. Note that (1.), (3.) are trivial to verify, based on the our assumption that  $x_k^T z_k = n\mu_k > \epsilon$ . Condition (2.) follows since we know that our iterates lie in  $\mathcal{N}_{\mu_k}(\zeta_k, \lambda_k)$ , while we have  $x_k^T z_k > \epsilon$ , by assumption. Indeed, from the neighbourhood conditions (19), we have that  $x_k^i z_k^i \geq \frac{\gamma\mu}{n} x_k^T z_k$ . Hence, there exists  $\bar{\epsilon} = \frac{\gamma\mu\epsilon}{n} > 0$  such that  $x_k^i z_k^i > \bar{\epsilon}$ , for all  $k \geq 0$ , for all  $i = \{1, \dots, n\}$ .

Finally, we have to show that the right hand side of (20) is uniformly bounded. To that end, we bound the right-hand side of the second block equation of (20) as follows:

$$\left\| Ax_k + \sigma_k \mu_k (y_k - \lambda_k) - b - \frac{\sigma_k \mu_k}{\mu_0} \bar{b} + \mu_k (y_k - \lambda_k - \frac{\bar{b}}{\mu_0}) - \mu_k (y_k - \lambda_k - \frac{\bar{b}}{\mu_0}) \right\|_2 \leq \frac{\mu_k}{\mu_0} \|\bar{b}\|_2 + \mu_k \left\| y_k - \lambda_k - \frac{\bar{b}}{\mu_0} \right\|_2,$$

where we used the neighbourhood conditions (19). Boundedness follows from Premise 1. A similar reasoning applies for bounding the right-hand side of the first block equation, while the right-hand side of the third block equation is bounded directly from the neighbourhood conditions. Combining the previous completes the proof.  $\square$

In the following Lemma, we prove by contradiction that the parameter  $\mu_k$  of Algorithm IP-PMM converges to zero, given that Premise 1 holds. The proof is based on the developments in [12,20] and is only partially given here, for ease of presentation.

**Lemma 8** *Given Premise 1, and a sequence  $(x_k, y_k, z_k) \in \mathcal{N}_{\mu_k}(\zeta_k, \lambda_k)$  produced by Algorithm IP-PMM, the sequence  $\{\mu_k\}$  converges to zero.*

*Proof* Assume, by virtue of contradiction, that  $\mu_k > \epsilon$ , for all  $k \geq 0$ . Then, we know that the Newton direction obtained by the algorithm at every iteration, after solving (20), will be uniformly bounded by a constant dependent only on  $n$ , that is, there exists a positive constant  $C^\dagger$ , such that  $\|(\Delta x_k, \Delta y_k, \Delta z_k)\|_2 \leq C^\dagger$ . As in Lemma 6, we define:

$$\tilde{r}_p(\alpha) = Ax_k(\alpha) + \mu_k(\alpha)(y_k(\alpha) - \lambda_k) - b - \frac{\mu_k(\alpha)}{\mu_0} \bar{b},$$

and

$$\tilde{r}_d(\alpha) = -Qx_k(\alpha) + A^T y_k(\alpha) + z_k(\alpha) - \mu_k(\alpha)(x_k(\alpha) - \zeta_k) - c - \frac{\mu_k(\alpha)}{\mu_0} \bar{c},$$

for which we know that equalities (42) and (43) hold, respectively. Take any  $k \geq 0$  and define the following functions:

$$\begin{aligned} f_1(\alpha) &:= (x_k + \alpha \Delta x_k)^T (z_k + \alpha \Delta z_k) - (1 - \alpha(1 - \frac{\sigma_{\min}}{2})) x_k^T z_k, \\ f_2^i(\alpha) &:= (x_k^i + \alpha \Delta x_k^i)(z_k^i + \alpha \Delta z_k^i) - \gamma_\mu \mu_k(\alpha), \quad i = 1, \dots, n, \\ f_3(\alpha) &:= (1 - 0.01\alpha) x_k^T z_k - (x_k(\alpha))^T (z_k(\alpha)), \\ g_2(\alpha) &:= \frac{\mu_k(\alpha)}{\mu_0} C_N - \|(\tilde{r}_p(\alpha), \tilde{r}_d(\alpha))\|_2, \end{aligned}$$

where  $\mu_k(\alpha) = \frac{(x_k + \alpha \Delta x_k)^T (z_k + \alpha \Delta z_k)}{n}$ ,  $(x_k(\alpha), y_k(\alpha), z_k(\alpha)) = (x_k + \alpha \Delta x_k, y_k + \alpha \Delta y_k, z_k + \alpha \Delta z_k)$ . We would like to show that there exists  $\alpha^* > 0$ , such that:

$$f_1(\alpha) \geq 0, \quad f_2^i(\alpha) \geq 0, \quad \text{for all } i = 1, \dots, n, \quad f_3(\alpha) \geq 0, \quad g_2(\alpha) \geq 0, \quad \text{for all } \alpha \in (0, \alpha^*].$$

These conditions model the requirement that the next iteration of Algorithm IP-PMM must lie in the updated neighbourhood:  $\mathcal{N}_{\mu_{k+1}}(\zeta_k, \lambda_k)$ . Note that Algorithm IP-PMM updates the parameters  $\lambda_k, \zeta_k$  only if the selected new iterate belongs to the new neighbourhood, defined using the updated parameters. Hence, it suffices to show that  $(x_{k+1}, y_{k+1}, z_{k+1}) \in \mathcal{N}_{\mu_{k+1}}(\zeta_k, \lambda_k)$ . It is important to observe that we do not require that the scaled infeasibilities are bounded with respect to the semi-norm defined in (17). This is because the aforementioned norm requires that  $\text{rank}(A) = m$ . In other words, we are using a slightly different neighbourhood, and that does not affect the global convergence of the method (but only its complexity).

Proving the existence of  $\alpha^* > 0$ , such that each of the aforementioned functions is positive, follows exactly the developments in Lemma 6, with the only difference being that the bounds on the directions are not explicitly specified in this case. Using the same methodology as in Lemma 6, while keeping in mind our assumption, namely  $x_k^T z_k > \epsilon$ , and hence  $x_k^i z_k^i > \bar{\epsilon}$ , we can show that:

$$\alpha^* := \min \left\{ 1, \frac{\sigma_{\min} \epsilon}{2(C^\dagger)^2}, \frac{(1 - \gamma_\mu) \sigma_{\min} \bar{\epsilon}}{2(C^\dagger)^2}, \frac{0.49 \epsilon}{2(C^\dagger)^2}, \frac{\sigma_{\min} C_N \epsilon}{2\mu_0(\xi_2)} \right\}, \quad (46)$$

where  $\xi_2$  is a bounded constant dependent on  $C^\dagger$ , and defined as in (44). However, using the inequality:

$$\mu_{k+1} \leq (1 - 0.01\alpha) \mu_k, \quad \text{for all } \alpha \in [0, \alpha^*],$$

we get that  $\mu_k \rightarrow 0$ , which contradicts our assumption that  $\mu_k > \epsilon$ , for all  $k \geq 0$ , and completes the proof.  $\square$

Finally, using the following Theorem, we derive a necessary condition for infeasibility.

**Theorem 4** *Given Premise 2, i.e. there does not exist a KKT triple for the pair (P)-(D), then Premise 1 fails to hold.*

*Proof* By virtue of contradiction, let Premise 1 hold. In Lemma 8, we proved that given Premise 1, Algorithm IP-PMM produces iterates that belong to the neighbourhood (19) and  $\mu_k \rightarrow 0$ . But from the neighbourhood conditions we can observe that:

$$\|Ax_k + \mu_k(y_k - \lambda_k) - b - \frac{\mu_k}{\mu_0}\bar{b}\|_2 \leq C_N \frac{\mu_k}{\mu_0},$$

and

$$\| -Qx_k + A^T y_k + z_k - \mu_k(x_k - \zeta_k) - c - \frac{\mu_k}{\mu_0}\bar{c} \|_2 \leq C_N \frac{\mu_k}{\mu_0}.$$

Hence, we can choose a sub-sequence  $\mathcal{K} \subseteq \mathbb{N}$ , for which:

$$\{Ax_k + \mu_k(y_k - \lambda_k) - b - \frac{\mu_k}{\mu_0}\bar{b}\}_{\mathcal{K}} \rightarrow 0, \text{ and } \{-Qx_k + A^T y_k + z_k - \mu_k(x_k - \zeta_k) - c - \frac{\mu_k}{\mu_0}\bar{c}\}_{\mathcal{K}} \rightarrow 0.$$

But since  $\|y_k - \lambda_k\|_2$  and  $\|x_k - \zeta_k\|_2$  are bounded, while  $\mu_k \rightarrow 0$ , we have that:

$$\{Ax_k - b\}_{\mathcal{K}} \rightarrow 0, \{c + Qx_k - A^T y_k - z_k\}_{\mathcal{K}} \rightarrow 0, \text{ and } \{x_k^T z_k\}_{\mathcal{K}} \rightarrow 0.$$

This contradicts Premise 2, i.e. that the pair (P)-(D) does not have a KKT triple, and completes the proof.  $\square$

In the previous Theorem, we proved that Premise 1 is a necessary condition for infeasibility, since otherwise, we arrive at a contradiction. Nevertheless, this does not mean that the condition is also sufficient. In order to obtain a more reliable test for infeasibility, that uses the previous result, we will have to use the properties of Algorithm IP-PMM. In particular, we can notice that if the primal-dual problem is infeasible, then the PMM sub-problem will stop being updated after a finite number of iterations. In that case, we know from Theorem 4 that the sequence  $\|(x_k - \zeta_k, y_k - \lambda_k)\|_2$  will grow unbounded. Hence, we can define a maximum number of iterations per PMM sub-problem, say  $K^\dagger \geq 0$ , as well as a very large constant  $C^\dagger \geq 0$ . Then, if  $\|(x_k - \zeta_k, y_k - \lambda_k)\|_2 > C^\dagger$  and  $k \geq K^\dagger$ , the algorithm is terminated. The specific choices for these constants will be given in the next section.

## 5 Computational Experience

In this section, we provide some implementation details and present computational results of the method, over a set of small to large scale linear and convex quadratic programming problems. The code was written in MATLAB and can be found in the following link:

<https://www.maths.ed.ac.uk/ERGO/software.html> (source link).

### 5.1 Implementation Details

Our implementation deviates from the theory, in order to gain some additional control, as well as computational efficiency. Nevertheless, the theory has served as a guideline to tune the code reliably. There are two major differences between the practical implementation of IP-PMM and its theoretical algorithmic counterpart. Firstly, our implementation uses different penalty parameters for the proximal terms and the logarithmic barrier term. In particular, we define a primal proximal penalty  $\rho$ , a dual proximal penalty  $\delta$  and the barrier parameter  $\mu$ . Using the previous, the PMM Lagrangian function in (14), at an arbitrary iteration  $k$  of the algorithm, becomes:

$$\mathcal{L}_{\mu_k, \delta_k, \rho_k}^{PMM}(x; \zeta_k, \lambda_k) = c^T x + \frac{1}{2} x^T Q x - \lambda_k^T (Ax - b) + \frac{1}{2\delta_k} \|Ax - b\|_2^2 + \frac{\rho_k}{2} \|x - \zeta_k\|_2^2,$$

and (15) is altered similarly. The second difference lies in the fact that we do not require the iterates of the method to lie in the neighbourhood defined in (19) in order to gain efficiency. In what follows, we provide further details concerning our implementation choices.



### 5.1.1 Free Variables

The method takes as input problems in the following form:

$$\min_x \left( c^T x + \frac{1}{2} x^T Q x \right), \quad \text{s.t. } Ax = b, \quad x^I \geq 0, \quad x^F \text{ free},$$

where  $I = \{1, \dots, n\} \setminus F$  is the set of indices indicating the non-negative variables. In particular, if a problem instance has only free variables, no logarithmic barrier is employed and the method reduces to a standard proximal method of multipliers. Of course in this case, the derived complexity result does not hold. Nevertheless, a global convergence result holds, as shown in [31]. In general, convex optimization problems with only equality constraints are usually easy to deal with, and the proposed algorithm behaves very well when solving such problems in practice.

### 5.1.2 Constraint Matrix Scaling

In the pre-processing stage, we check if the constraint matrix is well scaled, i.e if:

$$\left( \max_{i \in \{1, \dots, m\}, j \in \{1, \dots, n\}} (|A^{ij}|) < 10 \right) \wedge \left( \min_{i \in \{1, \dots, m\}, j \in \{1, \dots, n\} : |A^{ij}| > 0} (|A^{ij}|) > 0.1 \right).$$

If the previous is not satisfied, we apply geometric scaling in the rows of  $A$ , that is, we multiply each row of  $A$  by a scalar of the form:

$$d^i = \frac{1}{\sqrt{\max_{j \in \{1, \dots, n\}} (|A^{i,j}|) \cdot \min_{j \in \{1, \dots, n\} : |A^{i,j}| > 0} (|A^{i,j}|)}}, \quad \text{for all } i \in \{1, \dots, m\}.$$

However, for numerical stability, we find the largest integer  $p^i$ , such that  $2^{p^i} \leq d^i$  and we set  $\tilde{d}^i = 2^{p^i}$ , for all  $i \in \{1, \dots, m\}$ . Hence, the scaling factors are powers of two. Based on the IEEE representation of floating point numbers, multiplying by a power of 2 translates into an addition of this power to the exponent, without affecting the mantissa. This scaling technique is based on the one used within the GLPK solver (see [15]).

### 5.1.3 Starting Point, Newton-step Computation and Step-length

We use a starting point, based on the developments in [23]. To construct it, we try to solve the pair of problems (P)-(D), ignoring the non-negativity constraints.

$$\tilde{x} = A^T (AA^T)^{-1} b, \quad \tilde{y} = (AA^T)^{-1} A(c + Q\tilde{x}), \quad \tilde{z} = c - A^T \tilde{y} + Q\tilde{x}.$$

However, in order to ensure stability and efficiency, we regularize the matrix  $AA^T$  and employ the Preconditioned Conjugate Gradient (PCG) method to solve these systems (in order to avoid forming  $AA^T$ ). We use the classical Jacobi preconditioner to accelerate PCG, i.e.  $P = \text{diag}(AA^T) + \delta I$ , where  $\delta = 8$ , is set as the regularization parameter.

Then, to guarantee positivity and sufficient magnitude of  $x_I, z_I$ , we compute  $\delta_x = \max\{-1.5 \cdot \min(\tilde{x}^I, 0)\}$  and  $\delta_z = \max\{-1.5 \cdot \min(\tilde{z}^I, 0)\}$  and we obtain:

$$\tilde{\delta}_x = \delta_x + 0.5 \frac{(\tilde{x}^I + \delta_x e_{|I|})^T (\tilde{z}^I + \delta_z e_{|I|})}{\sum_{i=1}^{|I|} (\tilde{z}^I(i) + \delta_z)}, \quad \tilde{\delta}_z = \delta_z + 0.5 \frac{(\tilde{x}^I + \delta_x e_{|I|})^T (\tilde{z}^I + \delta_z e_{|I|})}{\sum_{i=1}^{|I|} (\tilde{x}^I(i) + \delta_x)},$$

where  $e_{|I|}$  is the vector of ones of appropriate dimension. Finally, we set:

$$y_0 = \tilde{y}, \quad z_0^I = \tilde{z}^I + \tilde{\delta}_z e_{|I|}, \quad z_0^F = 0, \quad x_0^I = \tilde{x}^I + \tilde{\delta}_x e_{|I|}, \quad x_0^F = \tilde{x}^F.$$

In order to find the Newton step, we employ a widely used predictor-corrector method. The practical implementation deviates from the theory at this point, in order to gain computational efficiency. We provide the algorithmic scheme in Algorithm PC for completeness, but the reader is referred to [23], for an extensive review of the method. It is important to notice that the centering parameter ( $\sigma > 0$ ) is not present in Algorithm PC. Solving two different systems, serves as a way of decreasing the infeasibility and allowing  $\mu_k$  (and hence  $\delta_k, \rho_k$ ) to decrease.

**Algorithm PC** Predictor-Corrector method

Compute the predictor:

$$\begin{bmatrix} -(Q + \Theta_k^{-1} + \rho_k I) & A^T \\ A & \delta_k I \end{bmatrix} \begin{bmatrix} \Delta_p x \\ \Delta_p y \end{bmatrix} = \begin{bmatrix} c + Qx_k - A^T y_k - \rho_k(x_k - \zeta_k) - d_{1_k} \\ b - Ax_k - \delta_k(y_k - \lambda_k) \end{bmatrix}, \quad (47)$$

where  $d_{1_k}^I = -\mu_k(X_k^I)^{-1}e_{|I|}$ ,  $d_{1_k}^F = 0$ , and  $(\Theta_k^I)^{-1} = (X_k^I)^{-1}(Z_k^I)$ ,  $(\Theta_k^F)^{-1} = 0$ .

Retrieve  $\Delta_p z$ :

$$\Delta_p z^I = d_{1_k}^I - (X_k^I)^{-1}(Z_k^I \Delta_p x^I), \quad \Delta_p z^F = 0.$$

Compute the step in the non-negativity orthant:

$$\alpha_x^{\max} = \min_{(\Delta_p x_k^{I(i)} < 0)} \left\{ 1, -\frac{x_k^{I(i)}}{\Delta_p x_k^{I(i)}} \right\}, \quad \alpha_z^{\max} = \min_{(\Delta_p z^{I(i)} < 0)} \left\{ 1, -\frac{z_k^{I(i)}}{\Delta_p z^{I(i)}} \right\},$$

for  $i = 1, \dots, |I|$ , and set:

$$\alpha_x = \tau \alpha_x^{\max}, \quad \alpha_z = \tau \alpha_z^{\max},$$

with  $\tau = 0.995$

▷ avoid going too close to the boundary.

Compute a centrality measure:

$$g_\alpha = (x^I + \alpha_x \Delta_p x^I)^T (z^I + \alpha_z \Delta_p z^I).$$

Set:  $\mu = \left( \frac{g_\alpha}{(x_k^I)^T z_k^I} \right)^2 \frac{g_\alpha}{|I|}$

Compute the corrector:

$$\begin{bmatrix} -(Q + \Theta_k^{-1} + \rho_k I) & A^T \\ A & \delta_k I \end{bmatrix} \begin{bmatrix} \Delta_c x \\ \Delta_c y \end{bmatrix} = \begin{bmatrix} d_{2_k} \\ 0 \end{bmatrix}, \quad (48)$$

with  $d_{2_k}^I = \mu(X_k^I)^{-1}e_{|I|} - (X_k^I)^{-1}\Delta_p X^I \Delta_p z^I$  and  $d_{2_k}^F = 0$ ,  $\Delta_p X = \text{diag}(\Delta_p x)$ .

Retrieve  $\Delta_c z$ :

$$\Delta_c z^I = d_{2_k}^I - (X_k^I)^{-1}(Z_k^I \Delta_c x^I), \quad \Delta_c z^F = 0.$$

$$(\Delta x, \Delta y, \Delta z) = (\Delta_p x + \Delta_c x, \Delta_p y + \Delta_c y, \Delta_p z + \Delta_c z).$$

Compute the step in the non-negativity orthant:

$$\alpha_x^{\max} = \min_{\Delta x^{I(i)} < 0} \left\{ 1, -\frac{x_k^{I(i)}}{\Delta x^{I(i)}} \right\}, \quad \alpha_z^{\max} = \min_{\Delta z^{I(i)} < 0} \left\{ 1, -\frac{z_k^{I(i)}}{\Delta z^{I(i)}} \right\},$$

and set:

$$\alpha_x = \tau \alpha_x^{\max}, \quad \alpha_z = \tau \alpha_z^{\max}.$$

Update:

$$(x_{k+1}, y_{k+1}, z_{k+1}) = (x_k + \alpha_x \Delta x, y_k + \alpha_z \Delta y, z_k + \alpha_z \Delta z).$$

We solve the systems (47) and (48), using the built-in MATLAB symmetric decomposition (i.e. `ldl`). Such a decomposition always exists, with  $D$  diagonal, for the aforementioned systems, since after introducing the regularization, the system matrices are guaranteed to be quasi-definite; a class of matrices known to be strongly factorizable, [34]. In order to exploit that, we change the default pivot threshold of `ldl` to a value slightly lower than the minimum allowed regularization value (`regthr`; specified in sub-section 5.1.4). Using such a small pivot threshold guarantees that no 2x2 pivots will be employed during the factorization process.

#### 5.1.4 PMM Parameters

At this point, we discuss how we update the PMM sub-problems in practice, as well as how we tune the penalty parameters (regularization parameters)  $\delta$ ,  $\rho$ . Notice that in Section 3 we set  $\delta_k = \rho_k = \mu_k$ . While this is beneficial in theory, as it gives us a reliable way of tuning the penalty parameters of the algorithm, it is not very beneficial in practice, as it does not allow us to control the regularization parameters in the cases of extreme ill-conditioning. Hence, we allow the use of different penalty parameters connected to the PMM sub-problems, while enforcing both parameters ( $\delta_k$ ,  $\rho_k$ ) to decrease at the same rate as  $\mu_k$  (based on the theoretical results in Sections 3, 4).

On the other hand, the algorithm is more optimistic in practice than it is in theory. In particular, we do not consider the semi-norm (17) of the infeasibility, while we allow the update of the estimates  $\lambda_k$ ,  $\zeta_k$ , to happen independently. In particular, the former is updated when the 2-norm of the primal infeasibility is sufficiently reduced, while the latter is updated based on the dual infeasibility.

More specifically, at the beginning of the optimization, we set:  $\delta_0 = 8$ ,  $\rho_0 = 8$ ,  $\lambda_0 = y_0$ ,  $\zeta_0 = x_0$ . Then, at the end of every iteration, we employ the algorithmic scheme given in Algorithm PEU. In order to ensure numerical stability, we do not allow  $\delta$  or  $\rho$  to become smaller than a suitable positive value,  $\text{reg}_{thr}$ . We set:  $\text{reg}_{thr} = \max\left\{\frac{\text{tol}}{\max\{\|A\|_\infty^2, \|Q\|_\infty^2\}}, 10^{-10}\right\}$ . This value is based on the developments in [29], in order to ensure that we introduce a controlled perturbation in the eigenvalues of the non-regularized linear system. The reader is referred to [29] for an extensive study on the subject. If the factorization fails, we increase the regularization parameters by a factor of 10 and repeat the factorization. If the factorization fails while either  $\delta$  or  $\rho$  have reached their minimum allowed value ( $\text{reg}_{thr}$ ), then we also increase this value by a factor of 10. If this occurs 5 consecutive times, the method is terminated with a message indicating ill-conditioning.

---

**Algorithm PEU** Penalty and Estimate Updates
 

---

```

 $r = \frac{|\mu_k - \mu_{k+1}|}{\mu_k}$  (rate of decrease of  $\mu$ ).
if ( $\|Ax_{k+1} - b\|_2 \leq 0.95 \cdot \|Ax_k - b\|_2$ ) then
     $\lambda_{k+1} = y_{k+1}$ ,
     $\delta_{k+1} = (1 - r) \cdot \delta_k$ .
else
     $\lambda_{k+1} = \lambda_k$ ,
     $\delta_{k+1} = (1 - \frac{1}{3}r) \cdot \delta_k$ , ▷ less aggressive is in this case.
end if
 $\delta_{k+1} = \max\{\delta_{k+1}, \text{reg}_{thr}\}$ , ▷ for numerical stability (ensure quasi-definiteness).
if ( $\|c + Qx_{k+1} - A^T y_{k+1} - z_{k+1}\|_2 \leq 0.95 \cdot \|c + Qx_k - A^T y_k - z_k\|_2$ ) then
     $\zeta_{k+1} = x_{k+1}$ .
     $\rho_{k+1} = (1 - r) \cdot \rho_k$ .
else
     $\zeta_{k+1} = \zeta_k$ .
     $\rho_{k+1} = (1 - \frac{1}{3}r) \cdot \rho_k$ .
end if
 $\rho_{k+1} = \max\{\rho_{k+1}, \text{reg}_{thr}\}$ .
  
```

---

### 5.1.5 Termination Criteria

There are four possible termination criteria. They are summarized in Algorithm TC. In the aforementioned algorithm,  $\text{tol}$  represents the error tolerance chosen by the user. Similarly,  $\text{IP}_{\max\text{it}}$  is the maximum allowed IPM iterations, also chosen by the user. On the other hand,  $\text{PMM}_{\max\text{it}}$  is a threshold indicating that the PMM sub-problem needs too many iterations before being updated (that is if  $k_{\text{PMM}} > \text{PMM}_{\max\text{it}}$ ). We set  $\text{PMM}_{\max\text{it}} = 5$ . When either  $\lambda_k$  or  $\zeta_k$  is updated, we set  $k_{\text{PMM}} = 0$ .

Let us now support the proposed infeasibility detection mechanism. In particular, notice that as long as the penalty parameters do not converge to zero, every PMM-subproblem must have a solution, even in the case of infeasibility. Hence, we expect convergence of the regularized primal (dual, respectively) infeasibility to zero, while from Section 4, we know that a necessary condition for infeasibility is that the sequence  $\|(x_k - \zeta_k, y_k - \lambda_k)\|_2$  diverges. If this behaviour is observed, while the PMM parameters  $\lambda_k$ ,  $\zeta_k$  are not updated (which is not expected to happen in the feasible but rank deficient case), then we can conclude that the problem under consideration is infeasible.

## 5.2 Numerical Results

At this point, we present the computational results obtained by solving a set of small to large scale linear and convex quadratic problems. In order to stress out the importance of regularization, we compare IP-PMM with a non-regularized IPM. The latter implementation, follows exactly from the implementation of IP-PMM, simply by fixing  $\delta$  and  $\rho$  to zero. In the non-regularized case, the minimum pivot of the `ldl` function is restored to its default value, in order to avoid numerical instability. Throughout all of the presented experiments, we set the number of maximum iterations to 200. It should be noted here, that we expect IP-PMM to require more iterations to converge, as compared to the non-regularized IPM. In turn, the Newton systems arising in IP-PMM, have better numerical properties (accelerating the factorization process), while overall the method is expected

**Algorithm TC** Termination Criteria

---

**Input:**  $\text{tol}$ ,  $k_{\text{IP}}$ ,  $k_{\text{PMM}}$ ,  $\text{IP}_{\text{maxit}}$ ,  $\text{PMM}_{\text{maxit}}$ .

**if**  $\left( \left( \frac{\|c - A^T y_k + Q x_k - z_k\|_2}{\max\{\|c\|_2, 1\}} \leq \text{tol} \right) \wedge \left( \frac{\|b - A x_k\|_2}{\max\{\|b\|_2, 1\}} \leq \text{tol} \right) \wedge (\mu_k \leq \text{tol}) \right)$  **then**

**return** Solution  $(x_k, y_k, z_k)$ .

**else if**  $\left( (\|c + Q x_k - A^T y_k - z_k + \rho_k(x_k - \zeta_k)\|_2 \leq \text{tol}) \wedge (\|x_k - \zeta_k\|_2 > 10^{10}) \right)$  **then**

**if**  $(k_{\text{PMM}} \geq \text{PMM}_{\text{maxit}})$  **then** ▷ PMM sub-problem not updated for many iterations

        Declare Infeasibility.

**end if**

**else if**  $\left( (\|b - A x_k - \delta_k(y_k - \lambda_k)\|_2 \leq \text{tol}) \wedge (\|y_k - \lambda_k\|_2 > 10^{10}) \right)$  **then**

**if**  $(k_{\text{PMM}} \geq \text{PMM}_{\text{maxit}})$  **then**

        Declare Infeasibility.

**end if**

**else if**  $(k_{\text{IP}} \geq \text{IP}_{\text{maxit}})$  **then** ▷ Maximum IPM iterations reached.

    No Convergence.

**end if**

---

to be significantly more stable. In what follows, we demonstrate that this increase in the number of iterations is benign, in that it does not make the resulting method inefficient. In contrast, we provide computational evidence that the acceleration of the factorization process more than compensates for the increase in the number of iterations. The experiments were conducted on a PC with a 2.2GHz Intel Core i7 processor (hexa-core), 16GB RAM, run under Windows 10 operating system. The MATLAB version used was R2018b.

### 5.2.1 Linear Programming Problems

Let us compare the proposed method with the respective non-regularized implementation, over the Netlib collection, [27]. The test set consists of 96 linear programming problems. We set the desired tolerance to  $\text{tol} = 10^{-6}$ . Firstly, we compare the two methods, without using the pre-solved version of the problem collection (e.g. allowing rank-deficient matrices). In this case, the non-regularized IPM converged for only 66 out of the total 96 problems. On the other hand, IP-PMM solved successfully the whole set, in 160 seconds (and a total of 2,609 IPM iterations). Hence, one of the benefits of regularization, that of alleviating rank deficiency of the constraint matrix, becomes immediately obvious.

However, in order to explore more potential benefits of regularization, we run the algorithm on a pre-solved Netlib library. In the pre-solved set, the non-regularized IPM converged for 93 out of 96 problems. The three failures occurred due to instability of the Newton system. The overall time spent was 353 seconds (and a total of 1,871 IPM iterations). On the other hand, IP-PMM solved the whole set in 161 seconds (and a total of 2,367 iterations). Two more benefits of regularization become obvious here. Firstly, we can observe that numerical stability can be a problem in a standard IPM, even if we ensure that the constraint matrices are of full row rank. Secondly, notice that despite the fact that IP-PMM required 26% more iterations, it still solved the whole set in 55% less CPU time. This is because in IP-PMM, only diagonal pivots are allowed during the factorization. We could enforce the same condition on the non-regularized IPM, but then significantly more problems would fail to converge (22/96) due to numerical instability.

In Table 1, we collect statistics from the runs of the two methods over some medium scale instances of the pre-solved Netlib test set.

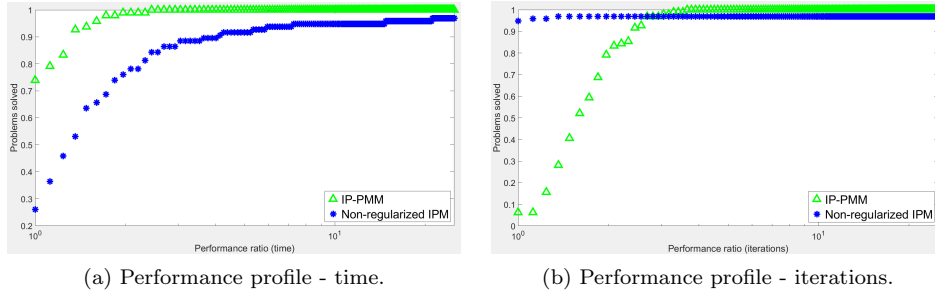
Table 1: Medium-Scale Netlib Problems

Name	$nnz(A)$	IP-PMM		IPM	
		Time (s)	IP-Iter.	Time (s)	IP-Iter.
80BAU3B	29,063	1.43	44	9.68	40
D6CUBE	43,888	1.26	25	9.64	22
DFL001	41,873	25.42	47	† <sup>1</sup>	†
FIT2D	138,018	8.52	27	23.94	25
FIT2P	60,784	1.24	24	1.56	19
PILOT87	73,804	7.21	49	95.04	46
QAP15	110,700	91.78	23	93.56	18

From Table 1, it becomes obvious that the factorization efficiency is significantly improved by the introduction of the regularization terms. In all of the presented instances, IP-PMM converged needing more iterations, but requiring less CPU time.

In order to summarize the comparison of the two methods, we include Figure 1, which contains the performance profiles, over the pre-solved Netlib set, of the two methods. IP-PMM is represented by the green line (consisted of triangles), while non-regularized IPM by the blue line (consisted of stars). In Figure 1a, we present the performance profile with respect to time required for convergence, while in Figure 1b, the performance profile with respect to the number of iterations. In both figures, the horizontal axis is in logarithmic scale, and it represents the ratio with respect to the best performance achieved by one of the two methods, for each problem. The vertical axis shows the percentage of problems solved by each method, for different values of the performance ratio. Robustness is “measured” by the maximum achievable percentage, while efficiency by the rate of increase of each of the lines (faster increase translates to better efficiency). For more information about performance profiles, we refer the reader to [10], where this benchmarking was proposed. One can see that all of our previous observations are verified in Figure 1.

Fig. 1: Performance profiles over the pre-solved Netlib test set.



### 5.2.2 Infeasible Problems

In order to assess the accuracy of the proposed infeasibility detection criteria, we attempt to solve 28 infeasible problems, coming from the Netlib infeasible collection ([27], see also [Infeasible Problems](#)). For 22 out of the 28 problems, the method was able to recognize that the problem under consideration is infeasible, and exit before the maximum number of iterations was reached. There were 4 problems, for which the method terminated after reaching the maximum number of iterations. For 1 problem the method was terminated early due to numerical instability. Finally, there was one problem for which the method converged to the least squares solution, which satisfied the optimality conditions for a tolerance of  $10^{-6}$ . Overall, IP-PMM run all 28 infeasible problems in 34 seconds (and a total of 1,813 IPM iterations). The proposed infeasibility detection mechanism had a 78% rate of success over the infeasible test set, while no feasible problem was misclassified as infeasible. A more accurate infeasibility detection mechanism could be possible, however, the proposed approach is easy to

<sup>1</sup> The symbol † means that the method did not converge.

implement and cheap from the computational point of view. Nevertheless, the interested reader is referred to [4, 26] and the references therein, for various other infeasibility detection methods.

### 5.2.3 Quadratic Programming Problems

Next, we present the comparison of the two methods over the Maros–Mészáros test set ([22]), which is comprised of 122 convex quadratic programming problems. Notice that in the previous experiments, we used the pre-solved version of the collection. However, we do not have a pre-solved version of this test set available. Since the focus of the paper is not on the pre-solve phase of convex problems, we present the comparison of the two methods over the set, without applying any pre-processing. As a consequence, non-regularized IPM fails to solve 27 out of the total 122 problems. However, only 11 out of 27 failed due to rank deficiency. The remaining 16 failures occurred due to numerical instability. On the contrary, IP-PMM solved the whole set successfully in 127 seconds (and a total of 3,014 iterations). As before, the required tolerance was set to  $10^{-6}$ .

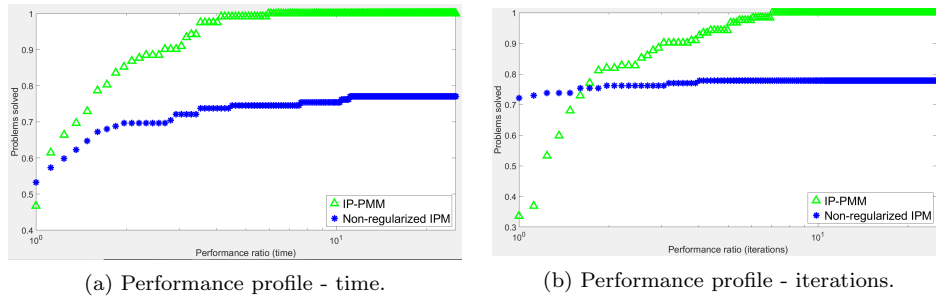
In Table 2, we collect statistics from the runs of the two methods over some medium scale instances of the Maros–Mészáros collection.

Table 2: Medium-Scale Maros–Mészáros Problems

Name	$nnz(A)$	$nnz(Q)$	IP-PMM		IPM	
			Time (s)	IP-Iter.	Time (s)	IP-Iter.
AUG2DCQP	40,000	40,400	4.70	83	7.21	111
CVXQP1L	14,998	69,968	25.54	38	†	†
CVXQP3L	22,497	69,968	45.69	59	†	†
LISWET1	30,000	10,002	1.07	30	1.86	40
POWELL20	20,000	10,000	1.26	30	1.61	25
QSHIP12L	16,170	122,433	0.91	23	†	†
STCQP1	13,338	49,109	0.38	16	6.89	13

In order to summarize the comparison of the two methods, we include Figure 2, which contains the performance profiles, over the Maros–Mészáros test set, of the two methods. IP-PMM is represented by the green line (consisted of triangles), while non-regularized IPM by the blue line (consisted of stars). In Figure 2a, we present the performance profile with respect to time required for convergence, while in Figure 2b, the performance profile with respect to the number of iterations.

Fig. 2: Performance profiles over the Maros–Mészáros test set.



Similar remarks can be made here, as those given when summarizing the linear programming experiments. One can readily observe the importance of the stability introduced by the regularization. On the other hand, the overhead in terms of number of iterations that is introduced due to regularization, is acceptable due to the acceleration of the factorization (since we are guaranteed to have a quasi-definite augmented system).

#### 5.2.4 Verification of the Theory

We have already presented the benefits of using regularization in interior point methods. A question arises, as to whether a regularized IPM can actually find an exact solution of the problem under consideration. Theoretically, we have proven this to be the case. However, in practice one is not allowed to decrease the regularization parameters indefinitely, since ill-conditioning will become a problem. Based on the theory of augmented Lagrangian methods, one knows that sufficiently small regularization parameters suffice for exactness (see [6], among others). In what follows, we provide a “table of robustness” of IP-PMM. We run the method over the Netlib and the Maros-Mészáros collections, for decreasing values of the required tolerance and report the number of problems that converged.

Table 3: Table of Robustness

Test Set	Tolerance	Problems Converged
Netlib (non-presolved)	$10^{-6}$	96/96
”	$10^{-8}$	95/96
”	$10^{-10}$	94/96
Netlib (presolved)	$10^{-6}$	96/96
”	$10^{-8}$	94/96
”	$10^{-10}$	94/96
Maros-Mészáros	$10^{-6}$	122/122
”	$10^{-8}$	121/122
”	$10^{-10}$	112/122

One can observe from Table 3 that IP-PMM is sufficiently robust. Even in the case where a 10 digit accurate solution is required, the method is able to maintain a success rate larger than 91%.

#### 5.2.5 Large Scale Problems

All of our previous experiments were conducted on small to medium scale linear and convex quadratic programming problems. We have shown (both theoretically and practically) that the proposed method is reliable. However, it is worth mentioning the limitations of the current approach. Since we employ exact factorization during the iterations of the IPM, we expect that the method will be limited in terms of the size of the problems it can solve. The main bottleneck arises from the factorization, which does not scale well in terms of processing time and memory requirements. In order to explore the limitations, in Table 4 we provide the statistics of the runs of the method over a small set of large scale problems. It contains the number of non-zeros of the constraint matrices, as well as the time needed to solve the problem. The tolerance used in these experiments was  $10^{-6}$ .

Table 4: Large-Scale Problems

Name	Collection	$nnz(A)$	time (s)	Status
fome21	Mittelman	751,365	567.26	opt
pds-10	Mittelman	250,307	40.00	opt
pds-30	Mittelman	564,988	447.81	opt
pds-60	Mittelman	1,320,986	2,265.39	opt
pds-100	Mittelman	1,953,757	-	no memory
rail582	Mittelman	402,290	91.10	opt
cre-b	Kennington	347,742	24.48	opt
cre-d	Kennington	323,287	23.49	opt
stocfor3	Kennington	72,721	4.56	opt
ken-18	Kennington	667,569	77.94	opt
osa-30	Kennington	604,488	1723.96	opt
nug-20	QAPLIB	304,800	386.12	opt
nug-30	QAPLIB	1,567,800	-	no memory

From Table 4, it can be observed that as the dimension of the problem grows, the time to convergence is significantly increased. This increase in time is disproportionate for some problems.



The reason for that, is that the required memory could exceed the available RAM, in which case the swap-file is activated. Access to the swap memory is extremely slow and hence the time could potentially increase disproportionately. On the other hand, we retrieve two failures due to lack of available memory. The previous issues could potentially be addressed by the use of iterative methods. Such methods, embedded in the IP-PMM framework, could significantly relax the memory as well as the processing requirements, at the expense of providing inexact directions. Combining IP-PMM (which is stable and reliable) with such an inexact scheme (which could accelerate the IPM iterations) seems to be a viable and competitive alternative and will be addressed in a future work.

## 6 Conclusions

In this paper, we present an algorithm suitable for solving convex quadratic programs. It arises from the combination of an infeasible interior point method with the proximal method of multipliers (IP-PMM). The method is interpreted as a primal-dual regularized IPM, and we prove that it is guaranteed to converge in a polynomial number of iterations, under standard assumptions. As the algorithm relies only on one penalty parameter, we use the well-known theory of IPMs to tune it. In particular, we treat this penalty as a barrier parameter, and hence the method is well-behaved independently of the problem under consideration. Additionally, we derive a necessary condition for infeasibility and use it to construct an infeasibility detection mechanism. The algorithm is implemented, and the reliability of the method is demonstrated. At the expense of some extra iterations, regularization improves the numerical properties of the interior point iterations. The increase in the number of iterations is benign, since factorization efficiency as well as stability is gained. Not only the method remains efficient, but it outperforms a similar non-regularized IPM scheme.

We observe the limitations of the current approach, due to the cost of factorization, and it is expected that embedding iterative methods in the current scheme might further improve the scalability of the algorithm at the expense of inexact directions. Since the reliability of IP-PMM is demonstrated, it only seems reasonable to allow for approximate Newton directions and still expect fast convergence. Hence, a future goal is to extend the theory as well as the implementation, in order to allow the use of iterative methods for solving the Newton system.

**Acknowledgements** SP acknowledges financial support from a Principal’s Career Development PhD scholarship at the University of Edinburgh, as well as a scholarship from A. G. Leventis Foundation. JG acknowledges support from the EPSRC grant EP/N019652/1. JG and SP acknowledge support from the Google Faculty Research Award: “Fast  $(1 + x)$ -order methods for linear programming”.

## References

- Altman, A., Gondzio, J.: Regularized Symmetric Indefinite Systems in Interior Point Methods for Linear and Quadratic Optimization. *Optim. Meth. and Soft.* **Vol. 11 & 12**, 275–302 (1999)
- Armand, P., Benoist, J.: Uniform Boundedness of the Inverse of a Jacobian Matrix Arising in Regularized Interior-Point Methods. *Math. Prog.* **Vol. 137** (No. 1&2), 587–592 (2013)
- Armand, P., Omheni, R.: A Mixed Logarithmic Barrier-Augmented Lagrangian Method for Nonlinear Optimization. *J. Optim. Theory and Appl.* **Vol. 173**, 523–547 (2017)
- Armand, P., Tran, N.N.: Rapid Infeasibility Detection in a Mixed Logarithmic Barrier-Augmented Lagrangian Method for Nonlinear Optimization. *Optim. Meth. and Soft.* **Vol. 34** (No. 5), 991–1013 (2019)
- Arreckx, S., Orban, D.: A Regularized Factorization-Free Method for Equality-Constrained Optimization. *SIAM J. Optim.* **Vol. 28** (No. 2), 1613–1639 (2018)
- Bertsekas, P.D.: *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific (1996)
- Bertsekas, P.D., Nedic, A., Ozdaglar, E.: *Convex Analysis and Optimization*. Athena Scientific (2003)
- Bertsekas, P.D., Tsitsiklis, N.J.: *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific (1997)
- Censor, Y., Zenios, A.: Proximal Minimization Algorithm with D-Functions. *J. of Optim. Theory and Appl.* **Vol. 73** (No. 3), 451–464 (1992)
- Dolan, D.E., Moré, J.J.: Benchmarking Optimization Software with Performance Profiles. *Math. Prog. Ser. A.* **Vol. 91**, 201–213 (2002)
- Eckstein, J.: Nonlinear Proximal Point Algorithms Using Bregman Functions, with Applications to Convex Programming. *Math. of Operational Research* **Vol. 18** (No. 1) (1993)
- Friedlander, M.P., Orban, D.: A Primal-Dual Regularized Interior-Point Method for Convex Quadratic Programs. *Math. Prog. Comp.* **Vol. 4** (No. 1), 71–107 (2012)
- Güler, O.: New Proximal Point Algorithms for Convex Minimization. *SIAM J. Optim.* **Vol. 2** (No. 4), 649–664 (1992)
- Gill, E.P., Robinson, P.D.: A Primal-Dual Augmented Lagrangian. *Comp. Optim. Appl.* **Vol. 51**, 1–25 (2012)

15. GLPK: <https://www.gnu.org/software/glpk/> (2018)
16. Gondzio, J.: Multiple Centrality Corrections in a Primal-Dual Method for Linear Programming. *Comp. Optim. and Appl.* **Vol. 6**, 137–156 (1996)
17. Gondzio, J.: Interior Point Methods 25 Years Later. *European J. of Operational Research* **Vol. 218** (No. 3), 587–601 (2013)
18. Hestenes, M.R.: Multiplier and Gradient Methods. *J. of Optim. Theory and Appl.* (No. 4), 303–320 (1969)
19. Iusem, A.N.: Some Properties of Generalized Proximal Point Methods for Quadratic and Linear Programming. *J. of Optim. Theory and Appl.* **Vol. 85** (No. 3), 593–612 (1995)
20. Kojima, M., Megiddo, N., Mizuno, S.: A Primal-Dual Infeasible-Interior-Point Algorithm for Linear Programming. *Math. Prog.* **Vol. 61**, 263–280 (1993)
21. Lan, G., Monteiro, R.D.C.: Iteration-Complexity of First-Order Augmented Lagrangian Methods for Convex Programming. *Math. Prog.* **Vol. 155** (No. 1-2), 511–547 (2016)
22. Maros, I., Mészáros, C.: A Repository of Convex Quadratic Programming Problems. *Optim. Meth. and Soft.* **Vol. 11 & 12**, 671–681 (1999)
23. Mehrotra, S.: On the Implementation of a Primal-Dual Interior-Point Method. *SIAM J. Optim.* **Vol. 2** (No. 4), 575–601 (1992)
24. Mizuno, S., Jarre, F.: Global and Polynomial-Time Convergence of an Infeasible-Interior-Point Algorithm using Inexact Computation. *Math. Prog.* **Vol. 84**, 105–122 (1999)
25. Nesterov, Y., Nemirovski, A.: Interior-Point Polynomial Algorithms in Convex Programming. *SIAM* (1993)
26. Nesterov, Y., Todd, M.J., Ye, Y.: Infeasible-Start Primal-Dual Methods and Infeasibility Detectors for Nonlinear Programming Problems. *Math. Prog.* **Vol. 82** (No. 2), 227–267 (1999)
27. Netlib: <http://netlib.org/lp> (2011)
28. Parikh, N., Boyd, S.: Proximal Algorithms. *Foundations and Trends in Optim.* **Vol. 3** (No. 1), 123–231 (2014)
29. Pougkakiotis, S., Gondzio, J.: Dynamic Non-diagonal Regularization in Interior Point Methods for Linear and Convex Quadratic Programming. *J. of Optim. Theory and Appl.* **Vol. 181** (No. 3), 905–945 (2019)
30. Powell, M.J.D.: A Method for Nonlinear Constraints in Minimization Problems. *Optim.* (No. 4), 283–298 (1969)
31. Rockafellar, R.T.: Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming. *Math. of Operations Research* **Vol. 1** (No. 2), 97–116 (1976)
32. Rockafellar, R.T.: Monotone Operators and the Proximal Point Algorithm. *SIAM J. Control and Optim.* **Vol. 14** (No. 5) (1976)
33. Rockafellar, R.T.: *Convex Analysis*. Princeton University Press (1996)
34. Vanderbei, R.: Symmetric Quasidefinite Matrices. *SIAM J. Optim.* **Vol. 5** (No. 1), 100–113 (1995)
35. Wang, Y., Fei, P.: An Infeasible Mizuno-Todd-Ye Type Algorithm for Convex Quadratic Programming with Polynomial Complexity. *Numer. Funct. Anal. and Optim.* **Vol. 28** (No. 3-4), 487–502 (2007)
36. Wright, S.J.: *Primal-Dual Interior Point Methods*. SIAM (1997)
37. Zhang, Y.: On the Convergence of a Class of Infeasible Interior-Point Methods for the Horizontal Linear Complementarity Problem. *SIAM J. Optim.* **Vol. 4** (No. 1), 208–227 (1994)
38. Zhou, G., Toh, K.C., Zhao, G.: Convergence Analysis of an Infeasible Interior Point Algorithm Based on a Regularized Central Path for Linear Complementarity Problems. *Comp. Optim. Appl.* **Vol. 27** (No. 3), 269–283 (2004)